REGIONAL TRANSPORTATION MONITORING INFORMATION SYSTEM

MONITORING AND PERFORMANCE ASSESSMENT SYSTEM (MAPAS)

for the

Southern California Association of Governments

System Specification Document

Version 1.0

September 2002

Prepared by:

200 E. Baker Street
Suite 205
Costa Mesa, CA 92626
Phone: (714) 513-1400
Fax: (714) 513-1407

Table of Contents

1. INTRODUCTION

The purpose of the *System Specification* document is to define the Regional Transportation Monitoring Information System's (RTMIS) architecture and functionality required to support business processes and manage system work flow. A functional requirements analysis has been completed and the system requirements have been documented in the RTMIS *Functional Specification* accepted by SCAG. The *System Specification* will serve as a blueprint to guide implementation of the envisioned system functionality described in the RTMIS *Functional Specification*.

2. GOALS AND OBJECTIVES

The goal of RTMIS is to improve the current business processes in place within SCAG and other regional transportation agencies for monitoring and analyzing the region's transportation system infrastructure and operating characteristics, and for assessing the performance of the system.

Objectives of RTMIS are:

- *To collect a variety of existing transportation and transportation related data sources.*

  RTMIS is not meant to replace any ongoing data collection efforts, but is intended to combine existing regional transportation and transportation related data resources together to provide a wealth of information to aid in understanding and solving Southern California's transportation issues.

- *To create a central data repository to house the different data sources.*

  The creation of a RTMIS central data repository will improve awareness and accessibility to regional transportation and transportation related data resources. As a result, RTMIS will assist in eliminating duplication of data collection efforts within agencies and the region as a whole.

- *To provide multiple end user data reporting, display and analysis tools.*

  The provision of reporting, display and analysis tools to the RTMIS user will reduce the amount of effort needed to identify, obtain, process, and report basic data and create associated products, such as maps and graphs, which are required for performance assessment, monitoring, and regional planning activities.

- *To provide regional accessibility to the system through the Internet.*

  The provision of RTMIS through the Internet will not only increase the accessibility of the greatly needed data and tools RTMIS has to offer to departments within SCAG, but will also allow the system data resources to be utilized and updated by other transportation agencies in the region.

- *To accommodate for future data and application growth of the system.*

As RTMIS matures, the functionality of the system will increase, as will the data content to support the new functionality. RTMIS growth has been accommodated in the design of the system.

3. SYSTEM OVERVIEW

Several important aspects of the RTMIS system design are discussed in the following pages. These sections introduce the RTMIS data, database design, hardware and third party software architecture, and the customized software component architecture.

3.1. RTMIS Data

RTMIS will rely on data sources both internal and external to SCAG for its database entities. The following table provides information on identified data sources to be included in RTMIS at deployment:

Table 3.1: RTMIS Data Sources

Data Set	Description
PeMS Loop Detector Locations	Inventory of Caltrans loop detector locations for the following counties: • Imperial, • Los Angeles, • Orange, • Riverside, • San Bernardino, and • Ventura Counties Produced by UC Berkeley, EECS Department.
PeMS Loop Detector Readings	Collections of pre-processed Caltrans loop detector readings for the associated loop detector locations; Produced by UC Berkeley, EECS Department; live data delivered in 5-minute aggregates.
HPMS	Highway inventory system; contains freeway characteristics for the state of California's highways and interstates; Produced by Caltrans, Division of TSI, Office of Highway Inventory and Performance; Delivered annually.
FUNC LRS	Linear referencing system of the state of California's highways and interstate infrastructure; Produced by Caltrans, Division of TSI, Office of Highway Inventory and Performance; Delivered annually.
Street Centerline	1:100,000 street centerline file of southern California's transportation network; Produced by Thomas Brothers Maps; Delivered annually.
OCTA MPAH	Orange County street segments representing 2001 Master Plan of Arterial Highways (MPAH) coverage, with traffic volumes, number of lanes and street names. Produced by OCTA; Delivered annually.
Highway and Major Street Inventory	Inventory of arterials in SCAG's sub-regions; Produced by SCAG's Information Services, Modeling Division; Updated every few years.
Census 2000	Census Places, Tracts and Blocks; Produced by the U.S. Census Bureau; Updated every decade/census survey.
Socio-Economic Projections	Current and projected figures by Census Tract for population, housing, employment, etc. figures; Produced by SCAG.
Land Use	Classification of land, based on how it is used, covering the SCAG region; Produced by SCAG in 1993.

It is planned that eventually RTMIS will contain a vast and comprehensive set of multi-modal transportation data. Priorities for incorporating data were established in the *RTMIS Functional Specification*. The following table illustrates which RTMIS data source(s) will fulfill the requested data element in year one implementation and the data component each data source satisfies. Other requested data elements not stated below will be added to RTMIS over time.

Table 3.2: Fulfillment of Requested Data Elements

Data Source	Data Component	Requested Data Elements
PeMS Loop Detector Readings	1	Loop Detector Locations
	2	Vehicle Miles Traveled
	2	Actual Speed
	2	Traffic Volume (Traffic Counts)
	2	Congestion Delay
Caltrans HPMS	1	Number of Lanes
	1	Surface Pavement Type
	1	Year of Surface Improvement
	1	Posted Speed
	1	Designated Truck Routes
	2	Percentage of Single Trucks
	2	Percentage of Combination
	1	Functional Classification
Caltrans FUNC LRS	1	Functional Classification
	1	Roadway Lengths
TBM Street Centerline	1	Roadway Lengths
OCTA MPAH	1	Number of Lanes
	2	Traffic Volume (ADT)
SCAG Highway and Major Street Inventory	1	Functional Classification
	1	Number of Lanes
	1	Posted Speed
	1	Truck Restrictions
Census 2000	3	Census Places
	3	Census Tracts
	3	Census Blocks
SCAG Socio-Economic Projections	3	Population
	3	Housing
	3	Employment
SCAG Land Use	3	Land Use
Unidentified	2	Traffic Incidents
	2	Truck/Goods Movement (Origin and Destination)
	2	Level of Service

RTMIS Data Component Category Assignment
1 - Transportation Infrastructure Inventory
2 - Transportation System Operational Characteristics
3 - Transportation Related Data

As presented by this table, some requested data elements are currently not being fulfilled by the data sets provided for RTMIS. These temporarily absent data elements are traffic incidents and truck/goods movement. Data sets providing this data will need to be identified[1] in order to fulfill the data requested and prioritized data elements.

[1] CHP and the TANN Incident data feed are potential sources for the traffic incident data.

3.2. RTMIS Database Design

RTMIS will have three logical databases: 1) the Production database; 2) the Staging database; and 3) the Pre-Staging database.

The Production database, also referred to as the data warehouse, will be the database that supports RTMIS user operations. This database will store historical data and will be multi-dimensional to support the more complex queries.

The Staging database will be a schema similar to the Production database, but it will not contain the historical data nor be multi-dimensional in design. The Staging database's main purposes are: 1) to ensure that data meets production standards before loading the data into the production database; 2) to prepare the data through transformation and standardization and 3) to optimize the system by preprocessing anticipated data queries and analysis. After data is successfully loaded in the Staging database and pre-processed, it will then be migrated into the Production database.

The Pre-Staging database will be the predecessor to the Staging database. The main purpose of the Pre-Staging database is to load the continuously updated collected data (*.txt files) into a RTMIS database before the data becomes unavailable[2]. The Pre-Staging database will initiate the process of loading the Internet acquired data into the pre-defined RTMIS database schema.

RTMIS data loading activities will be recorded in each database. Notifications will be delivered to system administrators for unsuccessful data loads.

3.3. PHYSICAL ARCHITECTURE and THIRD PARTY SOFTWARE COMPONENTS

RTMIS will be developed for a platform of Web-enabled GIS, database and software technologies, including:

- ESRI ArcIMS 4.0
- ESRI ArcInfo 8.2
- ESRI's ArcSDE 8.2[3]
- Microsoft's Internet Information Server (IIS) 5.0
- Microsoft's SQL Server 2000 Standard Edition
- Microsoft's Visual Studio .NET Enterprise Architect

RTMIS physical architecture will contain several dedicated servers and workstations, as well as utilize the services of other in-house server and workstation resources. (Refer to Figure 1.) RTMIS data will be acquired and loaded into the system in both an automated

[2] A PeMS data file is available on the EECS FTP site for approximately two hours. As new data becomes available, older data files on the FTP site are removed to provide space for the new data files.

[3] ESRI's ArcSDE is a highly recommended component in the system design. For an ArcSDE general software description, see Appendix A.

and semi-automated fashion. The data will be stored initially in an OLTP[4] database, the Pre-Staging and/or Staging databases, for pre-processing and then will be migrated into the OLAP[5] database, the data warehouse, for production use. The RTMIS Web Server will serve the data stored within the data warehouse to the Web. Either Microsoft's Internet Explorer 5.5 or higher or Netscape Communicator 4.75 or 6.0 Web browsers can be used to access the RTMIS application being hosted by the Web server and receive the system's services.

The 24/7 data collection and database loading will be solely managed by a RTMIS dedicated Windows 2000 Professional workstation, while the periodically updated data sets will be loaded in a semi-automated servic from existing workstations within SCAG. Pre-existing licenses within SCAG of ESRI's ArcInfo 8.2 must be installed on these workstations for the georeferencing and loading of spatial data.

The RTMIS database loading services will run on a RTMIS dedicated Windows 2000 Professional. In order to facilitate the loading of spatial data into the RTMIS databases, the RTMIS database loader workstation will need access to an existing license of ESRI's ArcInfo 8.2 license at SCAG during off-peak business hours. The Pre-Staging, Staging, and Production databases will reside on a RTMIS dedicated Windows 2000 Server, consisting of Microsoft's SQL Server, SQL Server's Analytical Services (OLAP) and Metadata Services, and ESRI's ArcSDE[6]. Presently, ESRI's ArcSDE is not a mandatory component of the RTMIS physical architecture design, however it should be considered for the following reasons:

- Supports the integration of spatial and attribute (tabular) data in a RDBMS;
- Manages the integrity of spatial data in a RDBMS;
- Enhances the performance of spatial data access;
- Supports database and schema portability; and
- Supports long transactions and versions.

The RTMIS Web Server will run on a dedicated Windows 2000 Internet Information Server (IIS) with ESRI's ArcIMS and Visual Studio .NET components. Visual Studio .NET components include Web services, data access services and custom components. The custom components will utilize the e-mail services of the existing SCAG mail server to send notifications to data set administrators and system users.

The diagram on the following page, Figure 3.3, illustrates the physical architecture and third party software component configuration.

[4] OLTP: Online Transaction Processing.
[5] OLAP: Online Analytical Processing. For a detailed description, see Appendix A.
[6] Should ESRI's ArcSDE become a required system component, it will be placed on the Production database server. For an ArcSDE software description, see Appendix A.

Figure 3.1: RTMIS Physical Architecture

SYSTEM FLOW

Data Placed on the Internet

1. Data transported via the Internet

Data Acquisition

2a. Data transported via the Internet.

2b. Data transported via manual process e.g. CD, manual FTP, etc.

2c. Data transported via manual process e.g. CD, copying over LAN, etc.

Initial RTMIS Data Upload

3a. Automated processes transmits data via the SCAG LAN from the Data Acquisition Workstation to Pre-Staging and then to the Staging database. The Database Loader workstation will facilitate this process.

3b. Semi-automated applications transmits data via the SCAG LAN from the Semi-Automated Data Loader Workstations to Staging database.

Data Warehouse Data Upload

4. Automated process transmits data from Staging to Production database. The Database Loader workstation will facilitate this process.

Data Warehouse/Web Server Connection

5. Data in the Production database is utilized by the RTMIS Web Server via the SCAG LAN.

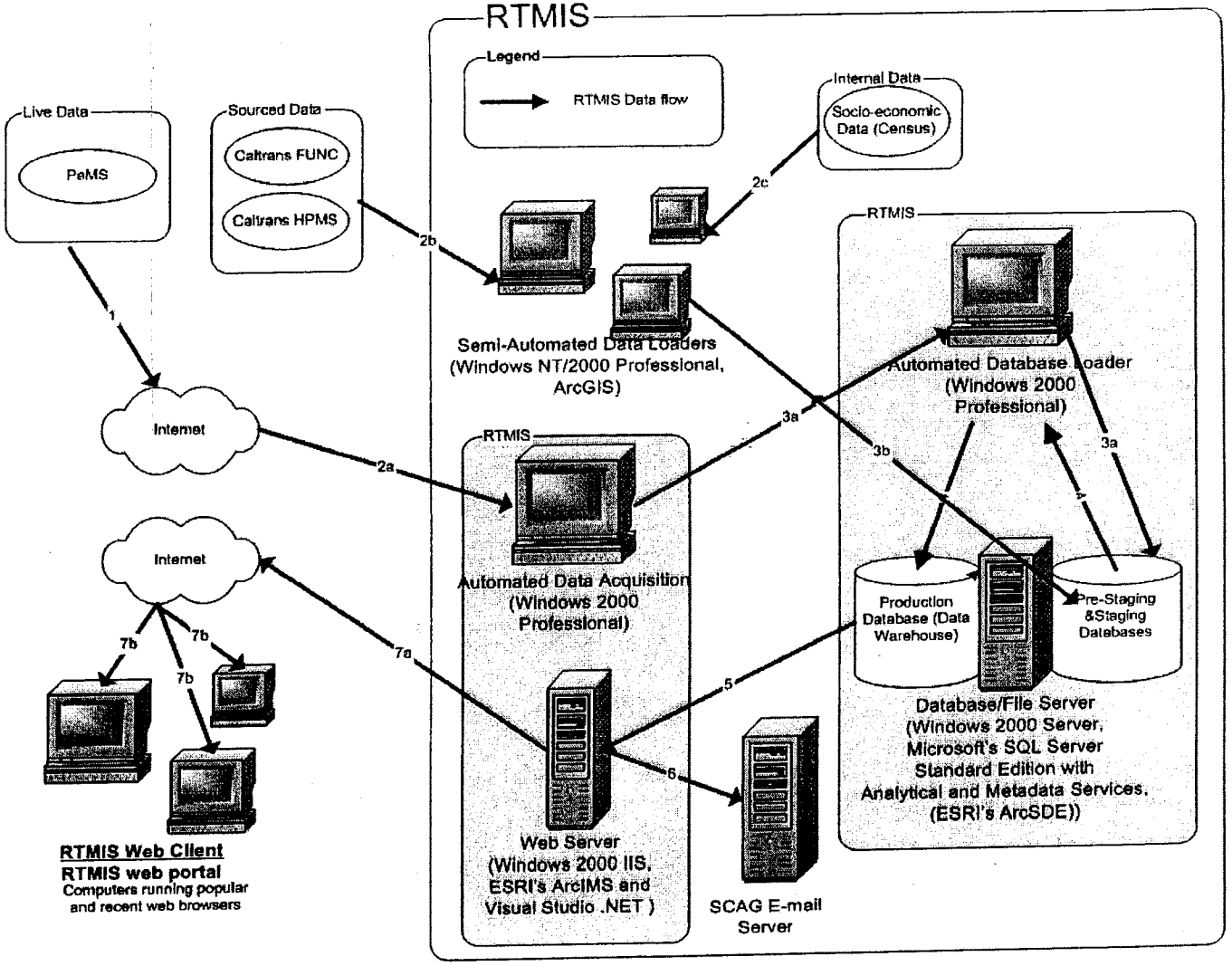
System Notification

6. The RTMIS access has access to the SCAG email server via the SCAG LAN.

RTMIS Access

7a. The RTMIS Web Server, hosted by SCAG, publishes its services on the Internet.

7b. Web browsers, clients of RTMIS, access the Web Server's services via the Internet.



The following figures provide the current status, purchased, not purchased or pre-existing, for each RTMIS software, operating system and hardware component.

Table 3.3: RMTIS Software Components

Software Components	Purchase Status	Description
ESRI's ArcIMS 4.0	Purchased 07/2002	1 Server License (includes 1 CPU) and 1 Addl. CPU License
ESRI's ArcInfo 8.2	Existing at SCAG	TBD
ESRI's ArcSDE 8.2	Not Purchased	1 Server License (includes 2 CPUs)
MS SQL Server 2000 Standard Edition	Purchased 07/2002	2 CPU Licenses
MS Visual Studio .NET Enterprise Architect	Purchased 07/2002	1 Developer License

Table 3.4: RTMIS Hardware Components

Hardware	Purchase Status	Description
• 2.0 GHz Dual Processor • 2 GB RAM • MS Windows 2000 Server • 3 x 73 GB RAID 5 Configured Hard Drives • Tape Backup Unit and Backup Software • Tape Media • Uninterrupted Power Supply	Not Purchased	Database Server
• 2.0 GHz Single Processor • 1 GB RAM • MS Windows 2000 Server with Internet Information Server (IIS) 5.0 • 73 GB Hard Drive • Uninterrupted Power Supply	Not Purchased	Web Server
• 2.0 GHz Single Processor • 512 MB RAM • MS Windows 2000 Professional	Not Purchased	Database Loader Workstation
• 2.0 GHz Single Processor • 256 MB RAM • MS Windows 2000 Professional • Uninterrupted Power Supply	Not Purchased	Data Acquisition Workstation

3.4. RTMIS SOFTWARE COMPONENTS DESIGN

The RTMIS software components will be distributed on a number of servers and workstations. A distributed system design is required to provide reasonable performance to the end user.

The RTMIS applications will be distributed over six areas:

1. *A Desktop Data Loading Application (Labeled as 'Application 1' in Figure 3.6):* RTMIS has a number of predetermined data sets that must be manually acquired and loaded into the system. These data sets will be uploaded periodically into the Staging database.

2. *An Automated Data Acquisition Service (Labeled as 'Service 1' in Figure 3.6):* Some RTMIS data sets will be continuously acquired for integration into the RTMIS system. This data will be acquired from known sources that are in known formats and reside on the Internet. These live data sources continually collect RTMIS relevant data and make the collected data available on the Internet. RTMIS will collect these data sets from the Internet at predetermined intervals 24/7. After the data is collected by this automated data acquisition service the data is loaded into the Pre-Staging database.

3. *An Automated Data Loading Service (Labeled as 'Service 2' in Figure 3.6):* After data has been loaded into the Staging database from semi-automated desktop application (#1) or into the Pre-Staging database by the automated service (#2) a second automated data loading service (#3) is responsible for loading, preparing and integrating the data and its associated metadata into the Production database. This data loading service will run at a predetermined interval.

4. *RTMIS Business Objects (Labeled as 'Business Components' in Figure 3.6):* The business components will support all operations performed by an RTMIS user or administrator using a Web interface. In Figure 3.6 these components reside on the same server as the RTMIS databases. These components will support the following RTMIS operations:

    - End User Analysis
    - Reporting
    - Mapping Services
    - Administrative Tasks

5. *RTMIS Web Application (Labeled as 'RTMIS Web application' in Figure 3.6):* The RTMIS Web application will reside on the Web server and will be responsible for managing end user and administrator user's interaction with the RTMIS functionality. Based on the user's actions on the Web client, a response will be generated for the user with support from the RTMIS business components.

6. *RTMIS Web Client Application (Labeled as 'RTMIS Web Client' in Figure 3.6):*

The RTMIS Web client will be a users' Web browser. The Web browsers that will be supported are listed in Section *3.3.*, "Physical Architecture and Third Party Software Components." The RTMIS application will be designed to minimize the amount of data downloaded into the client Web browser. This will increase performance for users with a low bandwidth connection to the Internet and minimize browser compatibility issues.

Figure 3.2: RTMIS Software Components

SYSTEM FLOW

Data Placed on the Internet

1. EECS Department (Univ. of Cal. At Berkeley) makes data available via FTP.

Data Acquisition

2a. Dedicated workstation acquires PeMS data 24/7.

2b. Sourced data is acquired from Caltrans through a manual process.

2c. SCAG internal data is acquired through a manual process.

Initial RTMIS Data Upload

3a. Automated processes running on the Database Loader workstation transmits data via the SCAG LAN from the Data Acquisition workstation to the Pre-Staging database and then to the Staging database.

3b. Semi-automated applications installed on existing SCAG department workstations load periodically update data into the Staging database.

Data Warehouse Data Upload

4. Data in the Staging database is standardized and migrated into the Production database, the data warehouse. The Database Loader workstation facilitates this process.

Data Warehouse/Web Server Connection

5. Data in the Production database is utilized by the RTMIS Web Server.

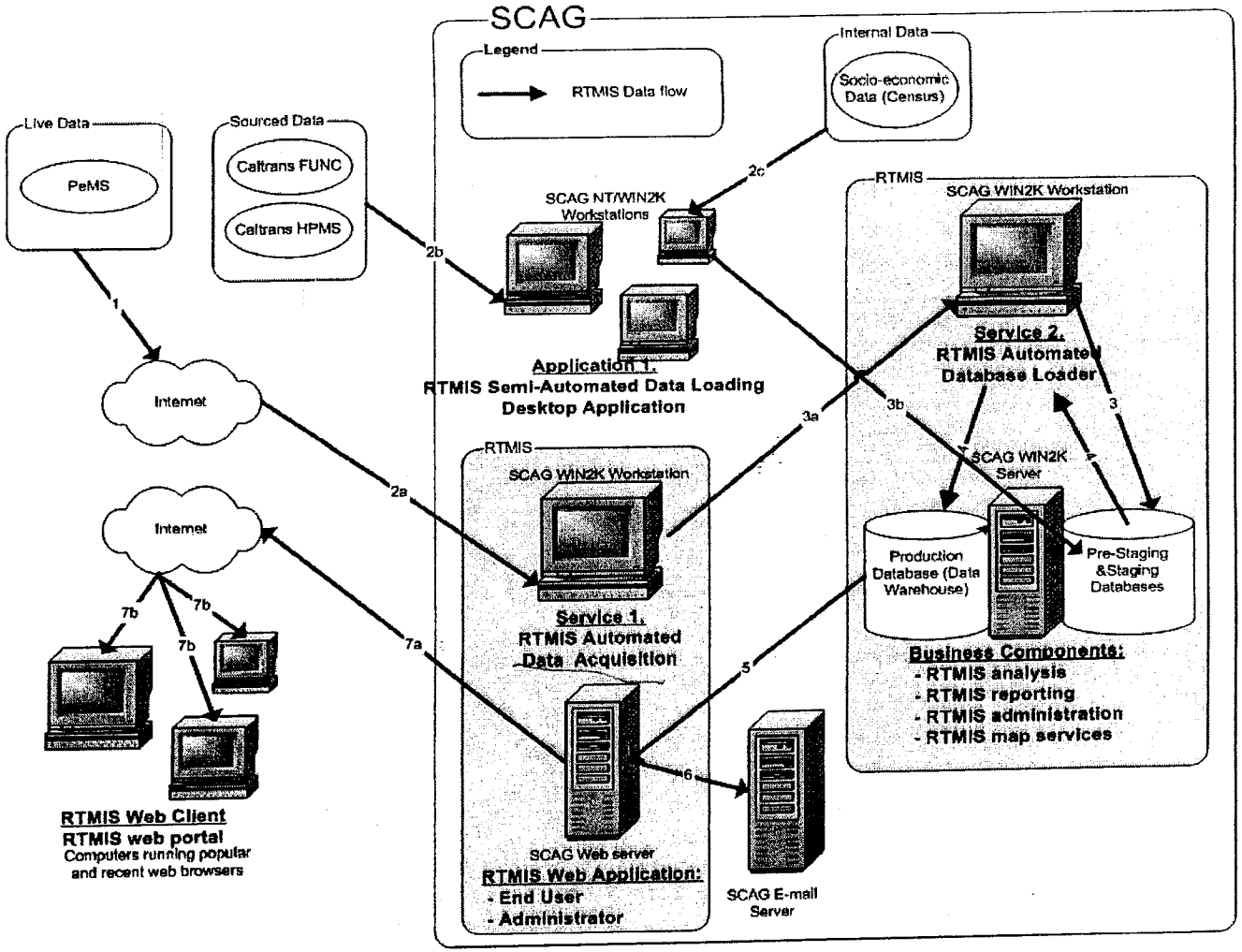
System Notification

6. SCAG's mail server is used to deliver system RTMIS analysis outputs to users and notifications to administrators.

RTMIS Access

7a. The RTMIS Web Server publishes its services on the Web.

7b. Web browsers, clients of RTMIS, access the Web Server's services.



4. SYSTEM PROCESS MODEL

4.1. DATA MODEL

Based on the data sources introduced in Section 3.1., "RTMIS Data", a broad overview of RTMIS data entities and their interrelations are described below. A more detailed document describing the RTMIS data model, the RTMIS *Data Definition*, will follow the RTMIS *System Specification*.

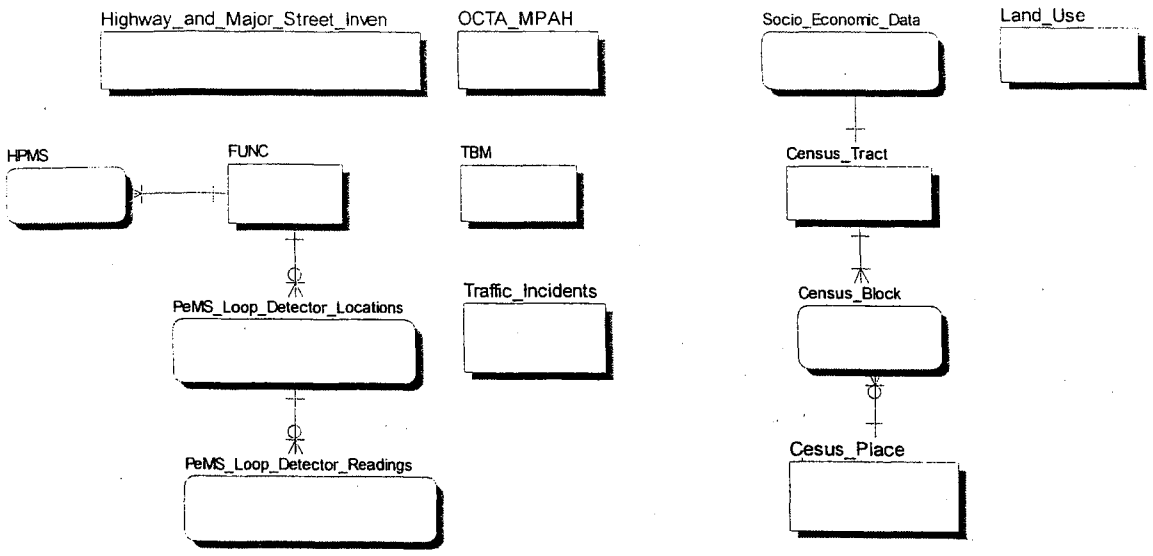
The creation of relationships between the different database entities adds value to any data model. By creating entity relationships between the database components, support for complex querying amongst multiple entities is provided, allowing users to discover underlying trends in the system. In all database systems the common way of interrelating different database entities is to identify a common attribute in the different database entities or to create one or more intermediate database entities to describe the relationship between different database entities.

Many of the RTMIS data entities, such as PeMS and HPMS, revolve around the FUNC LRS. The route IDs and absolute milepost values stored in the loop detector location table relate to the route IDs and measures stored in the FUNC LRS layer. The loop detector readings directly relate to the loop detector locations through the loop detector ID. As a result, traffic information provided by the loop detector readings can be referenced along the FUNC LRS by geo-referencing the loop detector locations to the FUNC LRS. In addition, HPMS relates to the FUNC LRS through the section IDs stored in both data entity tables.

It is unknown at this time how the base street network, proposed as Thomas Brothers Maps, the traffic incident data or other data will be related to the FUNC LRS. Figure 3, "Relationships of Entities Relevant to RTMIS", illustrates this disconnect between the "to be included" data entities and the central component, the FUNC LRS.

Socio-economic and land use data will be included in RTMIS. Although this data will not relate directly to the transportation layers, the data will be beneficial in supporting transportation queries and analysis. The Census 2000 spatial data layers are comprised of Census Tracts, Places and Blocks. A Census Tract is composed of many Census Blocks (one-to-many relationship), as a Census Place may be composed of many Census Blocks. The socio-economic data stored in a separate table from the spatial data relates explicitly with the Census Tract. A relationship does not exist between the socio-economic and land use data entities.

Figure 4.1: Relationships of Entities Relevant to RTMIS



4.2. BUSINESS PROCESSES

The general purpose of the RTMIS application and database system is to provide end users the ability to perform simple, analytical and comparative queries, obtain output from queries in a number of formats, and eventually edit selected attributes of some database entities. The RTMIS application will support representing data and queries in tabular, map and graph formats. To support end use functionality, a number of business processes will be developed to support the RTMIS application. Three major business processes have been identified: 1) data input into the system; 2) management of the system; and 3) end user interaction with the system.

4.2.1. Data Loading

RTMIS will not only host data from a variety of sources, but will in many cases standardize datasets and create relationships between these different data sources. Standardizing and interrelating datasets from a variety of sources is what will make meaningful queries spanning multiple datasets from multiple sources possible. Business processes will be in place to load, prepare, standardize, and interrelate datasets during the loading of data. A detailed description of current RTMIS data entities and their relationships are discussed in Section 4.1.

All RTMIS data loading could be classified in one of the following three categories:

Table 4.1: Data Loading Classification

Method/Interval	**Continuous:**	**Periodic:**
Automated:	Yes (Category 1)	Yes (Category 2)
Semi-automated:	No	Yes (Category 3)

1. *Continuous & Automated Data Load*
   Data will be loaded by services running on a workstation. The only user interaction required would be to stop or start the services. Typically data would be loaded 24 hours a day seven days a week (24/7). An example of this category of data would be loop detector readings.

2. *Periodic & Automated Data Load*
   This data will also be loaded by a services and therefore not require user interaction other than starting and stopping the service. This category of data will however be loaded at predetermined intervals and not continuously. An example of this category of data would be loop detector location data.

3. *Periodic & Semi-automated Data Load*
   This category of data will typically be loaded by a desktop application rather than a service. This desktop application would facilitate the loading of known data sets. User interaction would be required to invoke this application and guide the uploading of data into the RTMIS system. A prerequisite of this application is that these known data sets must be in a predetermined and published data format. Loading a data source that is in a predetermined format will substantially minimize the amount of user interaction during the loading process.

In broad terms data loading, regardless of above category, will involve the following sequential steps:

1. *Data Transformation*
   Data will be made available to the RTMIS database for uploading in a native format. For each data source this format will be predetermined and available to all interested parties. These exchange formats are discussed in detail in Section 4.2.1.1. Examples of possible native formats are ESRI Shapefiles and various text file formats. As part of the loading process, data will be transformed from the native format to the RTMIS Staging database's schema.

2. *Data Preparation and Standardization*
   Once data has been loaded and transformed into the Staging area the data needs to be prepared before it can be integrated into the Production database. The Production database will be a data warehouse for all previously loaded data. This phase of the data loading will accomplish all or some of the following tasks depending on the data set being loaded (See Section 4.2.1.1. of detail on operation performed on the various datasets):

   a. *Pre-calculation of Certain Database Fields*
      In order to optimize query response time on data in the data warehouse, some calculation of known attribute values will be done prior to loading the data into the data warehouse. Typically these values will be calculated from other known values from the same data set.
   b. *Standardization of Measures*
      Measure values may be acquired in different units for the same database entity attribute type. Before the data is loaded into the data warehouse, it will be converted to a common predefined data unit.

For example, all distances may be converted to feet, or entity coordinates may be derived from a common linear referencing system.

c. *Projection of Spatial Data to a Common Projection*
It is expected that spatial data will be acquired in different projections. Before the spatial data is stored in RTMIS, the data will be converted to a common predefined spatial projection. An example is converting a data set in UTM to State Plane.

d. *Building Relationships between Database Entities*
In order to perform complex queries involving multiple dataset, relationships need to be built or derived between different data sets. An examples of such an operation might be the association of a loop detector to a certain roadway segment, another example might be the association of a given roadway segment with a certain socio-economic attribute. Using these two examples a loop detector reading could be associated with socio-economic data.

3. *Data Integration*
Once data in the Staging database has successfully been prepared and standardized the data would be available for integration into the data warehouse. Since all data preparation is done prior to loading the data into the data warehouse, data integration from the Staging database to the data warehouse will simply be a matter of data loading.

4.2.1.1. Further Discussion of Data Transformation and Standardization

The following table presents the native format for each data source:

Table 4.2: Data Set File Formats

Data Set	Native File Format
PeMS Loop Detector Locations	Text File (*.txt)
PeMS Loop Detector Readings	Text File (*.txt)
Caltrans HPMS	Microsoft Access (*.mdb)
Caltrans FUNC	ESRI Shapefile (*.shp, *.dbf, *.shx) and Microsoft Access (*.mdb)
TBM Street Centerline	ESRI Shapefile (*.shp, *.dbf, *.shx)
OCTA MPAH	ESRI Coverage (.e00)
SCAG Highway and Major Street Inventory	ESRI Coverage (*.e00) & Microsoft Access (.dbf)
Census 2000	ESRI Shapefile (*.shp, *.dbf, *.shx)
SCAG Socio-Economic Projections	Microsoft Excel Spreadsheet (*.xls)
SCAG Land Use 1993	ESRI Shapefile (*.shp, *.dbf, *.shx)

As described in the "Business Processes, Data Loading" section, data sets will need to be transformed and standardized in the RTMIS Pre-Staging and/or Staging databases. The following table provides identified transformation or standardization processes that must be applied to each data entity before it is loaded into the RTMIS data warehouse.

Table 4.3: Data Set Transformation and Standardization

Data Set	Transformation and Standardization Process
PeMS Loop Detector Locations	• Extract data elements from native format. • Geo-reference spatial location (X,Ys) to FUNC LRS. • Reproject to common spatial projection.
PeMS Loop Detector Readings	• Extract of data elements from native format. • Pre-calculate database fields. • Convert to common units.
Caltrans HPMS	• Extract data elements from native format. • Convert to common units.
Caltrans FUNC	• Extract data elements from native format. (Microsoft Access database only.) • Reproject to common spatial projection. (Shapefile only.) • Conflate to TBM.
TBM Street Centerline	• Reproject to common spatial projection. • Conflate to FUNC LRS.
OCTA MPAH	• Reproject to common spatial projection. • Convert to common units.
SCAG Highway and Major Street Inventory	• Reproject to common spatial projection. • Geo-reference spatial location (X,Ys) to TBM Street Centerline. (Microsoft Access Database only.)
Census 2000	• Reproject to common spatial projection.
SCAG Socio-Economic Projections	• Extract data elements from native format.
SCAG Land Use	• Reproject to common spatial projection. • Convert to common units.

4.2.2. System Administration

Substantial parts of the RTMIS system will function in an automated manner. The RTMIS system will however required some periodic maintenance. RTMIS maintenance functionality will be exposed to a RTMIS administrator through a secure login. Depending on the type of maintenance task, the tasks will be performed using either a desktop or Web interface. Maintenance tasks will include:

- Administering RTMIS user accounts through a Web interface;
- Monitoring the status of past and present RTMIS automated tasks using a Web interface; and
- Loading certain datasets into the Staging database using a desktop application.

RTMIS system administrator will also receive notification from RTMIS when crucial automated tasks fail. Notification will be provided via email and possible wireless devices like pagers.

4.2.3. End User System Deliveries

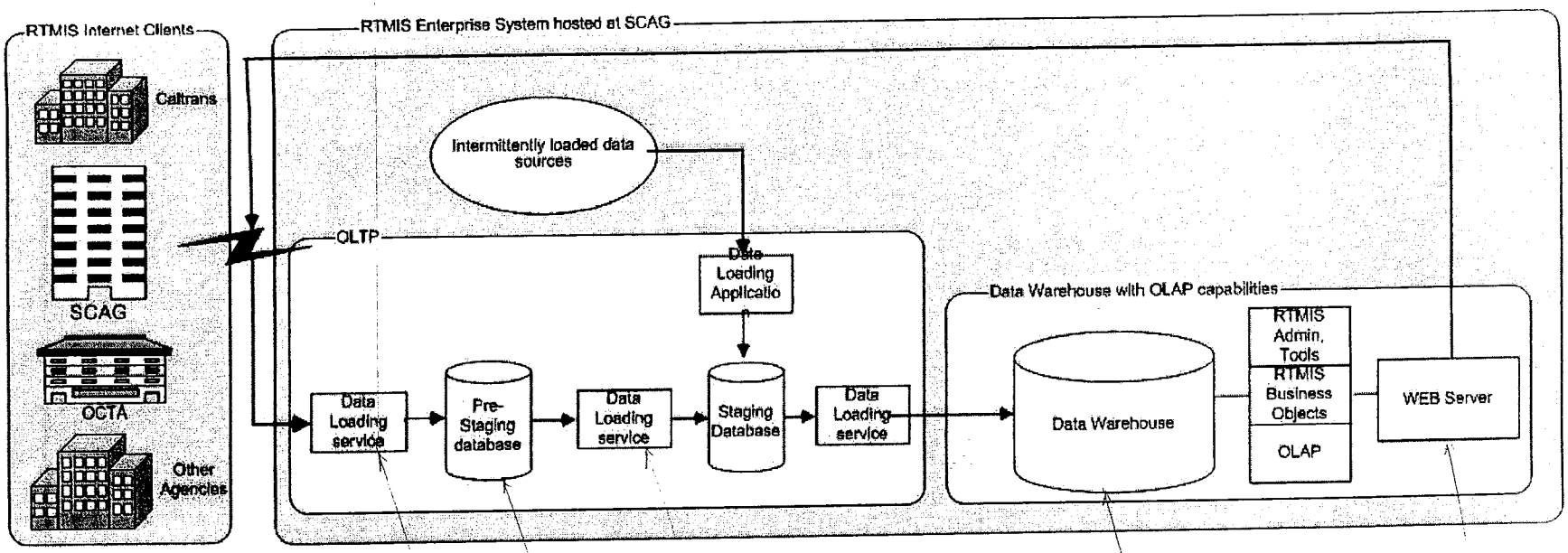
End users of the RTMIS system will access the RTMIS data warehouse through a number of Web interfaces. End user interaction will be substantial, but in broad terms, end user interaction can be summarized in the following categories:

- Perform queries on data in the RTMIS data warehouse. These queries will include standardized queries and customizable queries. These queries will also include simple queries on a single database entity or more complex queries spanning multiple related database entities. Queries will be optimized since data will be stored in a warehouse and online analytical processing (OLAP) capabilities will be utilized.
- RTMIS data and query results will be presented to the user using map displays, graphs and/or reports. These outputs will be delivered to the user as a Web page, email, or a file delivered via email or FTP.
- Users will be able to perform analysis on RTMIS datasets. Analysis will include:
    - Comparing two RTMIS data entities of the same type, e.g. attributes of two road segments.
    - Comparing two RTMIS data entity groups, e.g. loop detector readings from one day's morning rush hour versus the loop detector readings from the next day's morning rush hour.
    - Summarizing database entities that meet a user defined criteria, e.g. find all loop detector locations with a particular type of sensor or find all traffic volume values at a particular time.
    - Summarizing attribute values for a certain database entity based on another database entity with a spatial dimension, e.g. find all loop detectors on a given roadway segment or find all loop detectors that are in Tustin.
    - Analyzing Historical Trends, e.g. over a period of time, provide all actual speeds at a given time for a particular loop detector.

4.3. WORKFLOW

Generally speaking, RTMIS will facilitate the integration of a variety of data sets into the Production database. This Production database will be a data warehouse for all data loaded over time. Value will be added to each of these disparate datasets by transforming and standardizing each data set to a predefined format and by integrating these data sets with each other, as well as, with previously loaded data in the data warehouse. Integrated data sets will facilitate queries spanning these once discreet data sets. The envisioned flow of data through the RTMIS system is illustrated in Figure 4.2 and described in the following paragraphs.

Figure 4.2: Flow of data through the RTMIS system.



All data is initially loaded into an online transaction processing (OLTP) system, comprised of the Pre-Staging and Staging databases, before being loaded into the data warehouse. Data is transformed and standardized by the OLTP database and then loaded into the data warehouse. After the data is loaded into the data warehouse the end user can access the RTMIS data entities in the data warehouse for query and reporting purposes. Data is loaded into the OLTP database from two source types; the data is either acquired live from the Internet or from physical files that are intermittently transmitted and loaded.

Data from the Internet is acquired by an automated data acquisition service and loaded into a Pre-Staging database. This 24/7 automated data acquisition service is located on a dedicated SCAG RTMIS workstation that polls for new Internet data. As soon as the new data is detected by the automated data acquisition service, it is downloaded via the Internet and stored as a physical file on the workstation's hard drive. Periodically these physical files are loaded into the RTMIS Pre-Staging database's backlog table prior to being transformed into the RMTIS database schema[7]. After the files have been loaded into the Pre-Staging database, they are deleted from the workstation's hard drive.

In addition to the automated data acquisition service that loads Internet acquired data into the Pre-Staging database, RTMIS also acquires data that is more static in nature[8] using a semi-automated data loading desktop application. The desktop data loading application automatically performs any data transformation and integration requirements prior to loading the data into the Staging database. An authorized RTMIS system administrator is responsible for obtaining the more static data sets, ensuring they adhere to the published format requirements and loading the data into the Staging database by using the data loading application. Published format requirements include:

- The necessary file format,
- Associated field definitions,
- Field content description (e.g. units for measures), and
- Projection in the case of spatial data.

Although the loading of these data sets into RTMIS requires user input, user input will be minimized. The user input is limited to:

- Logging into the RTMIS data loading application,
- Identifying the RTMIS data entity type to be loaded,
- Pointing the application to the appropriate data source, and
- Entering basic metadata about the data set to be loaded.

After the data is loaded into the Pre-Staging database or the Staging database, the data sets are integrated with each other into the Production database. The integration is accomplished through an automated database loading service.

In the case of data loaded into the Staging database by the semi-automated data loading desktop application, minimal data preparation and standardization is required

[7] Data loaded into a database can be more easily analyzed and transformed than physical files simply residing on a workstation's hard drive.

[8] Examples of more static data sets are the socio-economic and transportation infrastructure data sets.

prior to integrating the data into the Production database since most of the preparation and standardization is performed during the data load into the Staging database. When this automated data loading service detects new data in the Staging database, it integrates the data and metadata into the Production database. After the data is successfully loaded into the Production database, the identical data residing in the Staging database is removed. Unless otherwise specified by the user of the desktop data loading application, data is only integrated into the Production database during off-peak RTMIS usage hours[9].

Data integration from the Pre-Staging database into the Production database requires a series of steps. The first step involves retrieving the physical files from the Pre-Staging database's backlog table and parsing the contents of the file into system's preliminary database schema in the Pre-Staging database. Next, the data is loaded from the Pre-Staging database into the Staging database at regular intervals throughout the day. During the Pre-Staging to Staging database conversion, all data preparation, standardization and removal of duplicate entries functions is performed. After the data is integrated with the other RTMIS data sets loaded externally into the Staging database, it is then loaded into the Production database. As to not degrade the performance of the Production database, the data in the Staging database will only be loaded into the Production database at off-peak hours. Once the temporary data is integrated into the Production database, it is removed from the Pre-Staging and Staging databases.

RTMIS automated database loading services require user input only when stopping and starting the service. Once a service is running no user input will be required. All RTMIS data loading services run independently from each other. In other words, if the service to load data into the Production database is not running, data can be loaded into the Staging database. As soon as the Production database data loading service becomes available again, data loaded into the Staging database during the service interruption is loaded into the Production database. System administrators can monitor tasks performed by the database loading services and are notified if/when a service encounters a problem.

Once the data is loaded into the Production database, the RTMIS data warehouse, an end user can access all the data loaded over time to perform queries and analysis on the data. The end user can access RTMIS end user functionality using one of the Web browsers stated in Section 3.3. via the Internet.

[9] RTMIS off-peak usage hours will be from 7 p.m. to 7 a.m. PST. Data integration into the RTMIS Production database will occur every day between 7 p.m. and 5 a.m. PST.

Figure 4.3: RTMIS Data Loading Frequency

Data Set	Loading Frequency
PeMS Loop Detector Readings	Continuous
PeMS Loop Detector Locations	Continuous
Caltrans FUNC LRS	Periodic
Caltrans HPMS	Periodic
TBM Street Centerline	Periodic
OCTA MPAH	Periodic
SCAG Highway and Major Streets Inventory	Periodic
Census 2000	Periodic
SCAG Socio-Economic Projections	Periodic
SCAG Land Use	Periodic

Figure 4.4: RTMIS Schedule of Tasks

RTMIS Tasks	Monday			Tuesday			Wednesday			Thursday			Friday			Saturday			Sunday		
Data Loading	✓			✓			✓			✓			✓			✓			✓			✓
Backup		✓			✓			✓			✓			✓			✓			✓	✓	
High User Availability			✓			✓			✓			✓			✓			✓				

Legend of Scheduled Times	
5:00 a.m. to 7:00 a.m.	
7:00 a.m. to 7:00 p.m.	
7:00 p.m. to 5 a.m.	
Task Performed	✓

5. SYSTEM FUNCTION MODEL

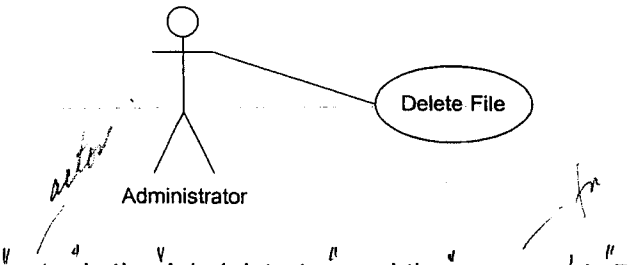
The purpose of this System Function Model is to define the functionality the user can expect to receive from the system in terms of business processes and work flow. These processes and workflows have been documented with use case scenarios. Use case analysis is an object-oriented system design methodology focusing on functional requirements. The objective of use case analysis is to identify discrete sets of interactions (known as a use case) between a system and its users, from which the user derives a certain value or achieves a specified goal.

Each use case describes one basic scenario, in which the goal of the user is achieved as expected, as well as other alternative scenarios, in which the same goal is reached through alternative sequences (i.e. user options), or the goal is abandoned (i.e. due to exceptional behavior, error handling, security breaches, etc.). The sum of all the use cases forms a complete functional view of a system from a perspective of the user. In other words, it presents a view of the system that defines who does what with the system, and for what purpose. After the functional requirements of the system have been captured in use cases, they can be used to guide the architecting of the system. They also help during implementation and testing to validate that the architecture supports the required functionality.

In its basic form, a use case diagram consists of the following notation:

- An actor (i.e. the Administrator of a system), which is depicted as a stick figure.
- A use case, which is depicted as an oval, and
- A communication between the two, which is depicted as a solid, straight line.

Thus, a use case in which a system administrator deletes a file would be depicted as follows:



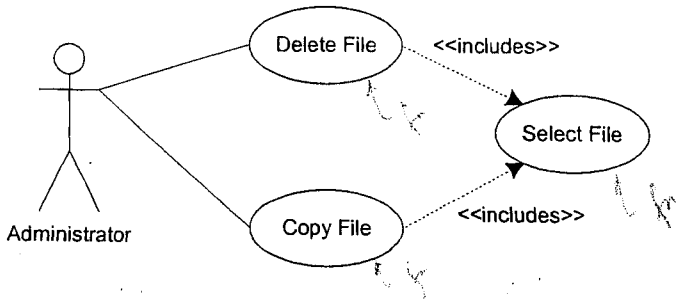
In this example, the actor is the Administrator, and the use case is Delete File. Note that although actors are usually human beings, they can also be other entities that interact with the system, such as an external system, or service.

Use case diagrams can be depicted with a wide range of detail. Often it is sufficient to diagram use cases at a very high-level, as in the example above. It is sometimes necessary to drill down to more detail, especially for the purposes of translating a functional specification into a more technical software specification. In these cases, it is helpful to break down the use cases into use cases that:

- Complement a primary use case in accomplishing its goal; and

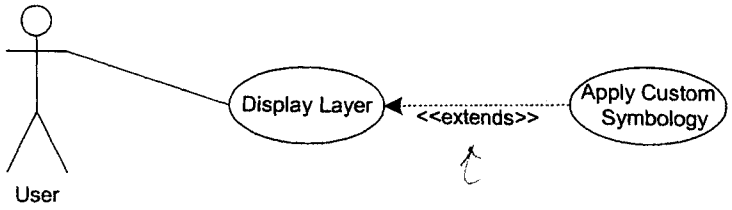
- Extend a use case to include optional variations.

When one use case complements another use case to help it achieve a goal, it is said that the complementary use case is "included" in the primary use case. This so-called include association is depicted by a dashed arrow that points to the included use case. For example, the use cases to delete a file or copy a file might be depicted as follows:



In this example, the use case to select a file is a required step in the use cases to delete a file and to copy a file. Since it is required by more than one use case, it is appropriate to separate out the Select File use case and establish it as a separate functionality.

Finally, it is useful to distinguish certain events that are not part of the basic course of events in a use case. Examples of these types of events are error handling situations or alternative courses that are associated with more than one calling use case. Since these types of events are not normally a part of a use case, and are not required for its successful completion, it is desirable to extract it from the use case and show it as a separate stand-alone use case that extends the functionality of the original use case. This type of relationship is called an extend association, and is depicted as a dashed arrow pointing to the use case being extended. For example, when displaying a spatial data layer in a GIS map display the user may decide to render the layer with custom symbology, instead of allowing the system to display the layer with the default symbology. This example would be depicted as follows:



Use case diagrams alone are not sufficient for conveying the system's functional specifications. In order to explain the primary goal of each use case, as well as the alternative scenarios which may take place in each use case, it is necessary to provide the use case description. According to industry standards, a use case description should generally have the following contents:

- *Use case name* – name of the use case as displayed in the use case diagram
- *Actor(s)* – the users of the system driving the functional value from the use case
- *Description* – A description of the goal or functional value the actor is expecting to achieve from the use case
- *Pre*-Condition(s) – prerequisite for the use case to be carried out

- *Initiation of the Use Case* – the situation that triggers the use case
- *Basic Course* – The basic set of steps that results in a successful outcome of the use case
- *Alternative Course(s)* – Additional steps that represent optional variations to the basic course of the use case
- *Post-Condition(s)* – The state of the system after the use case has completed
- *Issues* – Technical issues or other concepts impacting the use case that require further analysis or should be taken into consideration by the developer

The information presented in these sections is designed to provide an understanding of the scope of functionality being proposed for RTMIS, and to aid in the identification of functional requirements that have not been identified or satisfactorily addressed. When reviewing the diagrams, keep in mind that some requirements are not represented as separate use cases in the diagrams, but are captured within the contents of an individual use case description.

The use case analysis carried out for RTMIS has yielded a host of use cases that may be grouped into several functional areas, which are documented in sections 5.2 – 5.4. These are:

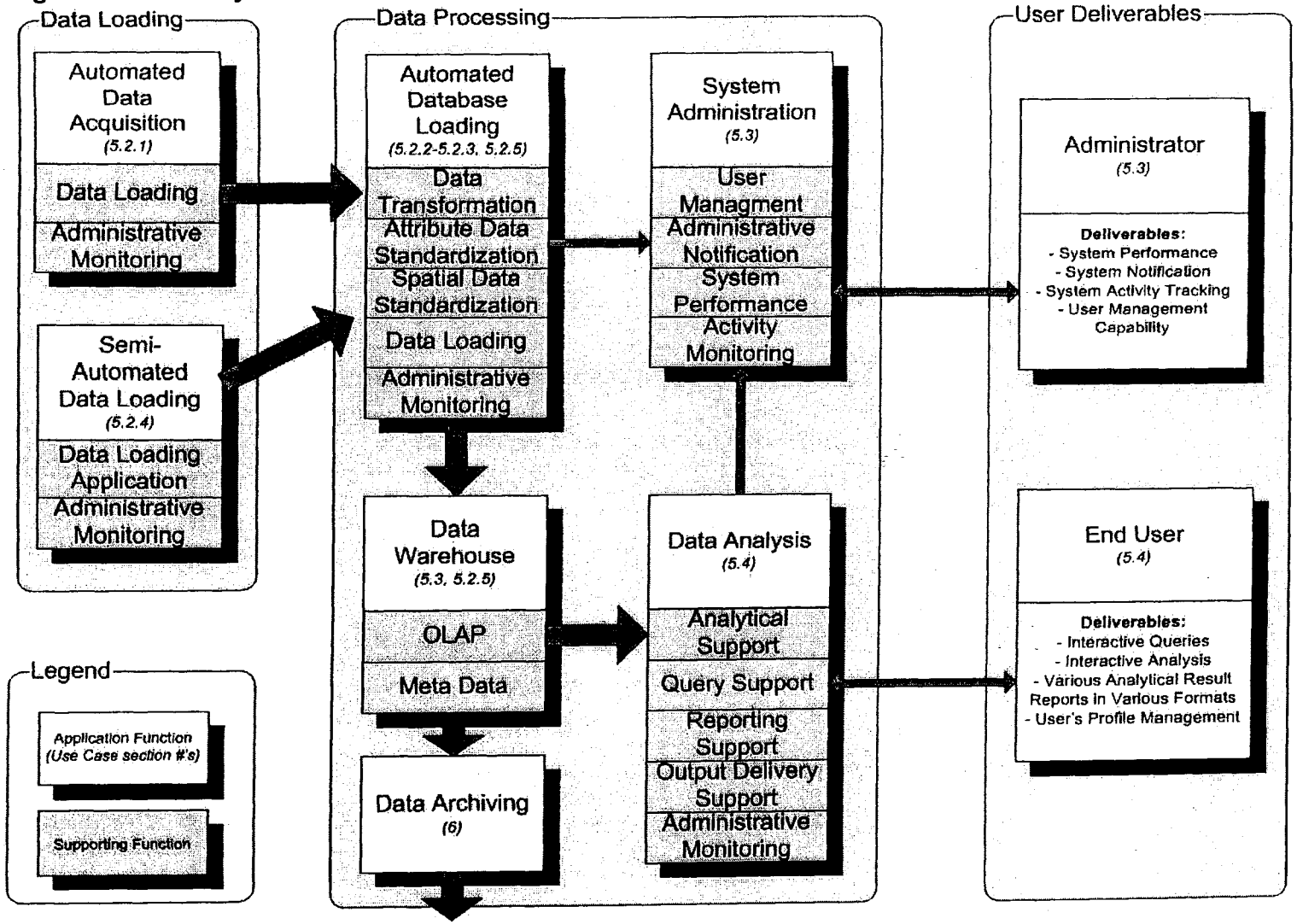
- Data Management
- Administrative Tools
- Analysis and Reporting Tools

Each of these sections provides a general description of the functional requirements associated with the functional area, a diagram depicting the use cases designed to capture those functional requirements, and detailed descriptions of each use case.

5.1 RTMIS System Overview

This diagram is not a UML diagram, but will serve as a reference as to how the following detailed UML diagrams will fit into the RTMIS system. Figure 5 illustrates the major and supporting functional areas of the RTMIS system. Throughout this diagram there will be reference to this document's use cases section numbers. These use case section numbers located in the diagram illustrate where the use cases fit into the RTMIS system. Note a section number can be referenced more than once in the diagram. The figure below provides and overview of the entire RTMIS system.

Figure 5: RTMIS System Overview



5.2. RTMIS Data Management

Loading data into the RTMIS data warehouse is a multi-stage process spanning multiple databases. The different data loading stages run parallel to each other as data is initially loaded, transformed, pre-processed and migrated between databases. For instance, as dynamically changing data is downloaded from the Internet into the Pre-Staging database; pre-processed data is being loaded into the RTMIS data warehouse.

The stages of the data loading process are listed below and then documented with use cases in the following subsections:

1. Internet Data to the Pre-Staging Database Backlog;
2. Pre-Staging Database Backlog to the Pre-Staging Database;
3. Pre-Staging Database to the Staging Database;
4. Periodically Updated Data to the Staging Database; and
5. Staging Database to the Production Database, the RTMIS data warehouse.

5.2.1. Internet Data to Pre-Staging Database Backlog

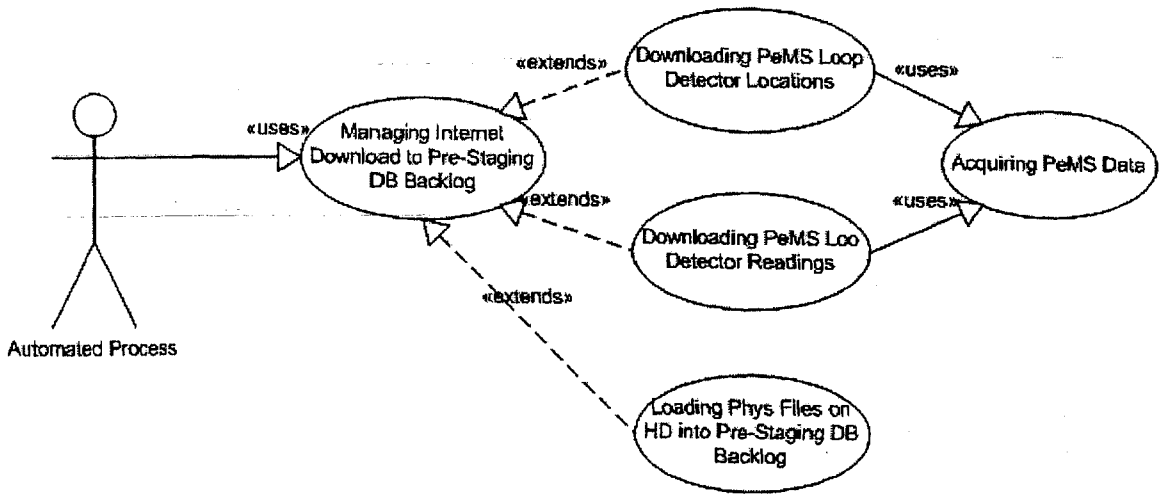
Five use cases define the process of loading Internet acquired data into the Pre-Staging database backlog. These use cases are:

1. *Managing Internet to Pre-Staging Database Backlog*
2. *Downloading PeMS Loop Detector Locations*
3. *Downloading PeMS Loop Detector Readings*
4. *Loading Physical File into the Pre-Staging Backlog*
5. *Acquiring PeMS Data*

"Managing Internet to Pre-Staging Database Backlog" use case directs the overall process of acquiring the more dynamic data from the Internet. "Downloading PeMS Loop Detector Locations" and "Downloading PeMS Loop Detector Readings" extends the capability of their parent use case, "Managing Internet to Pre-Staging Database Backlog." Each child use case is responsible for loading its Internet acquired data set into the Pre-Staging database backlog table. Both of these child use cases utilize the services of "Acquiring PeMS Data" use case to download its data from the Internet. During the download and database loading process by the two child use cases, if a connection can be made with the Pre-Staging database, then the downloaded Internet files are automatically stored in the Pre-Staging database backlog table. If the "Downloading PeMS Loop Detector Locations" or "Downloading PeMS Loop Detector Readings" use case can not connect to the Pre-Staging database, then the downloaded files remain temporarily on the downloading workstation's hard drive. The "Physical File to Pre-Staging Database Backlog" use case is responsible in loading these temporarily stored physical files on the workstation's hard drive to the Pre-Staging database backlog table.

Figure 6 illustrates the relationship between the "Internet to Pre-Staging Database Backlog" use cases.

Figure 6: Use Cases for Internet to Pre-Staging Backlog



5.2.1.1. Managing Internet Download to Pre-Staging Backlog[10]

ACTOR(S)

Automated Service

DESCRIPTION

Downloads the most recent PeMS data files and stores the downloaded files in the Pre-Staging database backlog table. If the connection to the Pre-Staging database can not be made, then the downloaded files will remain on the downloading workstation's hard drive. This use case determines if the data has been downloaded within a pre-determined amount of time or if any physical files exist on the hard drive.

PRECONDITON(S)

1. A time interval has been determined for loop detector location and reading data acquisition from the Internet.

INITIATION OF USE CASE

Continuous Service

BASIC COURSE

1. Has PeMS loop detector locations been downloaded within the pre-determined time period?
   a. If no, request "Downloading PeMS Loop Detector Locations" use case.
2. Has PeMS loop detector readings been downloaded within the pre-determined time period?
   a. If no, request "Downloading PeMS Loop Detector Readings" use case.
3. Are there any physical files on the hard drive?
   a. If yes, request "Physical File to Backlog" use case. (This request will only load one file at a time.)
4. Return to Step 1.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

Most recent PeMS data is downloaded and stored in the Pre-Staging database backlog.

---

[10] The Pre-Staging database backlog is a table within the Pre-Staging database that stores the downloaded text files (*.txt).

ISSUES

1. PeMS Sensor data is made available every 5 minutes and is available for about two hours after the file is initially made available.
2. PeMS Location data is available via Website download and is updated only when new Caltrans loop detector locations are added.

5.2.1.2. Downloading PeMS Loop Detector Locations

ACTOR(S)

Automated Service

DESCRIPTION

Downloads the PeMS loop detector locations *.txt to a physical file from the Internet to the SCAG data acquisition workstation and then attempts to add the physical file to the Pre-Staging backlog table.

PRECONDITON(S)

None

INITIATION OF USE CASE

The parent use case, "Managing Internet to Pre-Staging Backlog", requests the download of PeMS loop detector locations.

BASIC COURSE

1. Connect to Pre-Staging database.
   a. Create an action in the database action log.
   b. Log an activity for the action.
2. Request PeMS Location data from the child use case, "Acquiring PeMS Data".
3. Add physical file to Pre-Staging backlog table.
   a. If unable to connect from Step 1, go to "Alternative Course."
4. Remove the physical file from the hard drive.
5. Log activity for the action.
6. Close the action in the database action log.
7. Disconnect from the Pre-Staging database.

ALTERNATIVE COURSE

1. Abort.

POST-CONDITION(S)

PeMS loop detector locations *.txt is stored in the Pre-Staging backlog table.

ISSUES

None

5.2.1.3. Downloading PeMS Loop Detector Readings

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to download the PeMS loop detector readings *.txt from EECS Department's FTP site to a physical file on the SCAG data acquisition workstation and then attempts to add the physical file to the Pre-Staging database's backlog table.

PRECONDITON(S)

None

INITIATION OF USE CASE

The "Managing Internet to Backlog" use case requests the download of PeMS loop detector readings.

BASIC COURSE

1. Connect to Pre-Staging database.
   a. If able to connect, create action and log activity.
2. Request PeMS Sensor data from "Acquiring PeMS Data".
3. Add physical file to Pre-Staging backlog table.
   a. If not connected from Step 1, go to *Alternative Course.*
4. Remove the physical file from the local hard drive.
5. Log activity.
6. Close action.
7. Disconnect from Pre-Staging database.

ALTERNATIVE COURSE

1. Log activity.
2. Abort.

POST-CONDITION(S)

PeMS loop detector readings *.txt is stored in the Pre-Staging backlog table.

ISSUES

None

5.2.1.4. Loading Physical Files on the Hard Drive into Pre-Staging Database Backlog

ACTOR(S)

Automated Service

DESCRIPTION

Loads a physical file stored on the hard drive into the Pre-Staging database backlog table. If a connection to the Pre-Staging database could not be made during the initial download process, the physical files remain on the downloading workstation's hard drive until a connection to the database can be made. This use case makes another attempt to load these files into the Pre-Staging database backlog table.

PRECONDITION(S)

1. A physical file is stored on the hard drive that was not initially loaded into the Pre-Staging database backlog table.

INITIATION OF USE CASE

The "Managing Internet to Pre-Staging Backlog" use case requests to store a physical file on the hard drive in the Pre-Staging database backlog table.

BASIC COURSE

1. Connect to Pre-Staging database.
   a. If unable to connect, go to *Alternative Course*.
2. Create action and log activity.
3. Request the oldest physical file to be stored on the hard drive.
4. Add the physical file to the Pre-Staging database backlog.
5. Log activity.
6. Remove the physical file from the hard drive.
7. Log activity.
8. Close action.
9. Disconnect from the Pre-Staging database.

ALTERNATIVE COURSE

1. Log activity.
2. Abort.

POST-CONDITION(S)

A physical file is stored in the Pre-Staging database backlog.

ISSUES

None

5.2.1.4. Acquiring PeMS Data

ACTOR(S)

Automated Service

DESCRIPTION

Enables the system to obtain PeMS loop detector data.

PRECONDITON(S)

1. The EECS Department's FTP and Web sites at University of California, at Berkeley are available with data.
2. A successful connection and login to the sites with pre-assigned login and password.

INITIATION OF USE CASE

The "Downloading PeMS Loop Detector Locations" or "Downloading PeMS Loop Detector Readings" use case requests PeMS data acquisition.

BASIC COURSE

1. If the parent use case is connected to the Pre-Staging database, go to *Alternative Course "Connected to Pre-Staging Database."*
2. Connect to the EECS Department's FTP or Web service.
   a. If connection fails, go to *Alternative Course "Failed Connection to Website."*
3. Provide user login and password to service.
   a. If login fails, go to *Alternative Course "Failed Login."*
4. Request data file from service.
5. Download data file to physical drive.
   a. If download fails, go to *Alternative Course "Failed Download."*
6. If the parent use case is connected to the Pre-Staging database, go to *Alternative Course "Connected to Pre-Staging Database."*
7. Terminate connection with the FTP or Web service.
8. Make physical file name available to parent use case.

ALTERNATIVE COURSE

Connected to Pre-Staging Database:
1. Log activity.

Failed Connection to Website:
1. Abort.

Failed Login:
1. Abort.

Failed Download:
1. Abort.

POST-CONDITION(S)

1. The Actor has acquired the most recent PeMS data.
2. The physical file name is made available to the parent use case.

ISSUES

None

5.2.2. Pre-Staging Backlog to the Pre-Staging Database

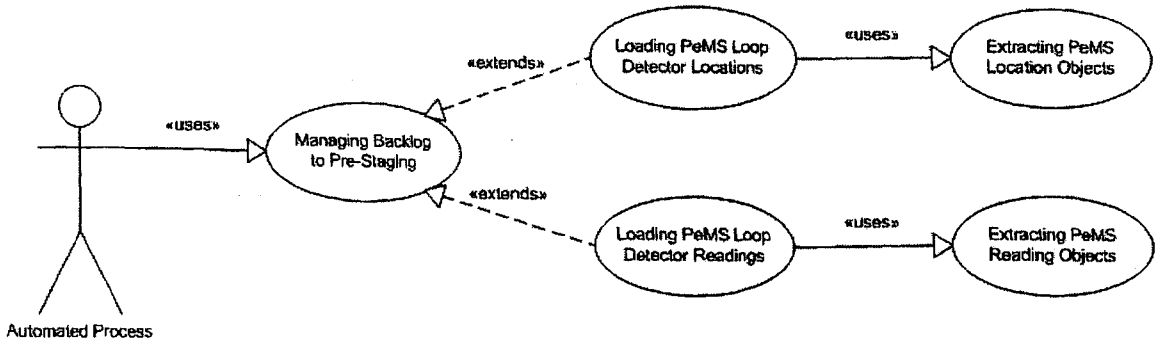
Five use cases define the process of extracting and loading data acquired from the Internet and stored in the Pre-Staging database backlog into the Pre-Staging database. These use cases are:

1. *Managing Pre-Staging Backlog to Pre-Staging Database*
2. *Loading PeMS Loop Detector Locations*
3. *Extracting PeMS Location Objects*
4. *Loading PeMS Loop Detector Readings*
5. *Extracting PeMS Reading Objects*

The "Managing Pre-Staging Backlog to the Pre-Staging Database" use case directs the overall process of loading the Internet acquired data into the Pre-Staging database. "Loading PeMS Loop Detector Locations" and "Loading PeMS Loop Detector Readings" use cases extend the capability of their parent use case, "Managing Pre-Staging Backlog to Pre-Staging Database." Each child use case is responsible for loading its data type into the appropriate Pre-Staging database data entity table. To accomplish this task, the child use cases utilize the services of its respective extractor use case. "Extracting PeMS Location Objects" use case extracts the loop detector location objects from the bar (|) delimited loop detector locations text file (*.txt). "Extracting PeMS Reading Objects" use case extracts the loop detector readings from the comma (,) delimited loop detector readings text file (*.txt).

Figure 7 illustrates the relationships between the "Pre-Staging Backlog to the Pre-Staging Database" use cases.

Figure 7: Use Cases for Pre-Staging Backlog to Pre-Staging Database



5.2.2.1. Managing Pre-Staging Backlog to the Pre-Staging Database

ACTOR(S)

Automated Service

DESCRIPTION

Loads data files from the Pre-Staging database's backlog table into the main Pre-Staging database tables.

PRECONDITON(S)

None

INITIATION OF USE CASE

A data file is stored in the Pre-Staging database backlog table that needs to be loaded into the main Pre-Staging database tables.

BASIC COURSE

1. Retrieve the oldest file from the Pre-Staging database backlog.
2. Identify the type.
   a. If the data type is PeMS loop detector locations, then request "Loading PeMS Loop Detector Locations" use case.
   b. If the data type is PeMS loop detector readings, then request "Loading PeMS Loop Detector Readings" use case.
3. If a physical file exists in the Pre-Staging database backlog table, return to Step 1.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

A Pre-Staging database backlog file has been identified as a PeMS loop detector location or reading file and the appropriate child use case to load the data into the main Pre-Staging database tables has been called.

ISSUES

None

5.2.2.2. Loading PeMS Loop Detector Locations into the Pre-Staging Database Table

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to load a PeMS loop detector locations *.txt from the Pre-Staging database backlog into the Pre-Staging database loop detector locations table.

PRECONDITON(S)

1. A PeMS loop detector locations *.txt exists in the Pre-Staging database backlog.

INITIATION OF USE CASE

The "Managing Backlog to Pre-Staging Database" use case identifies a PeMS loop detector locations *.txt in the Pre-Staging database backlog.

BASIC COURSE

1. Connect to Pre-Staging database.
   a. If unable to connect, go to *Alternative Course.*
2. Log activity.
3. Retrieve the PeMS loop detector locations *.txt from the Pre-Staging database backlog.
4. Request the "Extracting PeMS Loop Detector Location Objects" use case.
5. Retrieve a loop detector location object from the loop detector location collection.
6. Add the loop detector location object (Primary Keys – loop detector ID and activity ID) to the Pre-Staging database loop detector locations table.
7. Repeat Step 5 and 6 until all loop detector location objects have been added to the table.
8. Remove *.txt from the Pre-Staging database backlog.
9. Log activity.
10. Close connection with Pre-Staging database.

ALTERNATIVE COURSE

Unable to Connect:
1. Abort.

]

POST-CONDITION(S)

Data contained in the PeMS loop detector location *.txt from Pre-Staging database backlog table has been loaded into the Pre-Staging database loop detector locations table and the physical file removed from the Pre-Staging database backlog table.

ISSUES

1. It is assumed that the loop detector ID is a unique identifier for a physical loop detector and will remain valid over time.

5.2.2.3. Loading PeMS Loop Detector Readings into Pre-Staging Database Table

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to load a PeMS loop detector readings *.txt from the Pre-Staging database backlog into the Pre-Staging database loop detector readings table.

PRECONDITON(S)

1. A PeMS loop detector readings *.txt is in the Pre-Staging database backlog.

INITIATION OF USE CASE

The "Managing Backlog to Pre-Staging Database" use case identifies a PeMS loop detector readings *.txt in the Pre-Staging database backlog.

BASIC COURSE

1. Connect to Pre-Staging database.
   a. If unable to connect, go to *Alternative Course.*
2. Log activity.
3. Retrieve the PeMS loop detector readings *.txt from the backlog.
4. Request the "Extracting PeMS Loop Detector Reading Objects" use case.
5. Retrieve a loop detector reading object from the loop detector reading collection.
6. Verify that the loop detector ID for the loop detector reading object is stored in the loop detector locations table in the Pre-Staging database.
   a. If the loop detector ID is not found, go to *Alternative Course.*
7. Add loop detector reading object (Primary Keys – loop detector ID and activity ID) to the loop detector readings table (Pre-Staging).
8. Repeat Step 5 - 7 until all loop detector reading objects in the collection have been added to the table.
9. Remove *.txt from the Pre-Staging database backlog.
10. Log activity.
11. Close connection with Pre-Staging database.

ALTERNATIVE COURSE

Unable to Connect:
1. Abort.

Unmatched Sensor Reading:
1. Log activity.

2. Add the loop detector reading object to the Pre-Staging unmatched loop detector readings table.
3. Return to *Basic Course* (Step 8).

POST-CONDITION(S)

Data contained in the PeMS loop detector readings *.txt from Pre-Staging database backlog table has been loaded into a Pre-Staging database loop detector readings table the physical file removed from the Pre-Staging database backlog table.

ISSUES

1. It is assumed that the loop detector ID is a unique identifier for a physical loop detector and will remain valid over time.

5.2.2.4. Extracting PeMS Loop Detector Location Objects

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to extract the PeMS loop detector locations from the *.txt.

PRECONDITON(S)

1. PeMS Location *.txt exists in Pre-Staging backlog.

INITIATION OF USE CASE

"Loading PeMS Loop Detector Locations" use case requests the loop detector locations stored within the PeMS loop detector locations *.txt.

BASIC COURSE

1. Receive a PeMS loop detector locations *.txt.
2. Convert all locations from the bar (|) delimited *.txt into a collection of location objects.
   a. If the *.txt content cannot be converted, go to *Alternative Course.*

ALTERNATIVE COURSE

1. Log activity.
2. Abort.

POST-CONDITION(S)

The bar (|) delimited loop detector location content has been converted to a collection of loop detector location objects.

ISSUES

None

5.2.2.5. Extracting PeMS Loop Detector Reading Objects

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to extract the PeMS loop detector readings from the *.txt.

PRECONDITON(S)

PeMS 5-minute aggregate *.txt exists in the Pre-Staging backlog.

INITIATION OF USE CASE

"Loading PeMS Loop Detector Readings" use case requests the sensor readings stored within the PeMS *.txt.

BASIC COURSE

1. Receive PeMS loop detector readings *.txt.
2. Convert loop detector readings from the comma (,) delimited *.txt into a collection of loop detector reading objects.
   a. If the *.txt content cannot be converted, go to *Alternative Course.*

ALTERNATIVE COURSE

1. Log activity.
2. Abort.

POST-CONDITION(S)

The comma (,) delimited loop detector readings content has been converted to a collection of loop detector reading objects.

ISSUES

None

5.2.3. Pre-Staging Database to the Staging Database

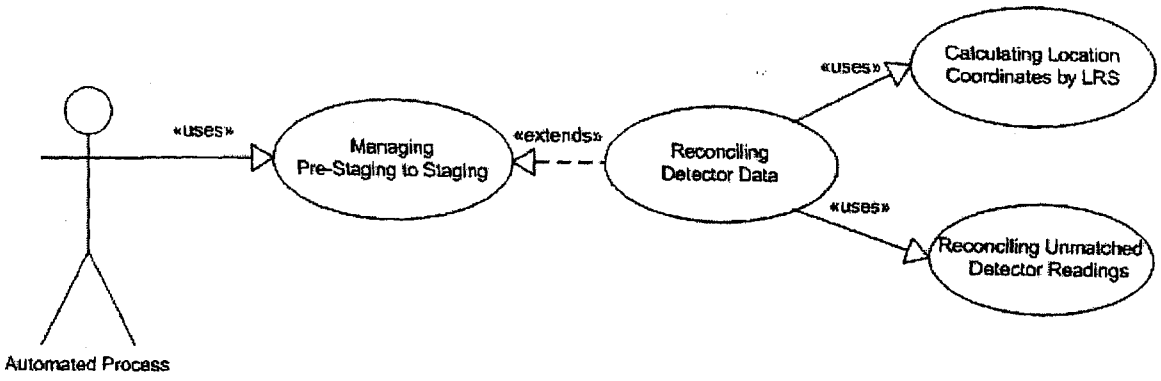
Four use cases define the process of loading data from the Pre-Staging database into the Staging database. These use cases are:

1. *Managing Pre-Staging Database to Staging Database*
2. *Reconciling Detector Data*
3. *Calculating Location Coordinated by LRS*
4. *Reconciling Unmatched Loop Detector Readings*

"Managing Pre-Staging Database to Staging Database" use case directs the overall process of loading data entities from the Pre-Staging database into the Staging database. "Reconciling Detector Data" use case extends the capability of its parent use case, "Managing Pre-Staging Database to Staging Database." "Reconciling Detector Data" use case reconciles loop detector data between the Pre-Staging database and the Staging database with the assistance of "Calculating Location Coordinates by LRS" and "Reconciling Unmatched Detector Readings" use cases. "Calculating Location Coordinates by LRS" use case is responsible for geo-referencing the loop detector locations along the linear referencing system. "Reconciling Unmatched Detector Readings" use case is responsible for matching unmatched loop detector readings to loop detector locations that enter the system after the readings.

Figure 8 illustrates the relationships between the "Pre-Staging to the Staging Database" use cases.

Figure 8: Use Cases for Data Loading from Pre-Staging to Staging Database



5.2.3.1. Managing Pre-Staging Database to Staging Database

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to reconcile and load newly collected data that in the Pre-Staging database into the Staging database.

PRECONDITON(S)

None

INITIATION OF USE CASE

This use case will active at all time.

BASIC COURSE

1. If detector location and reading data in the Pre-Staging database has NOT been reconciled with the detector location and reading data in the Staging database within the last predetermined amount of time call the "Reconciling Detector Data Staging Database" use case.
2. Return to Step 1.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

Data loaded into the Pre-Staging database is loaded into the Staging database.

ISSUES

None

5.2.3.2. Reconciling Detector Data with the Staging Database

ACTOR(S)

Automated Service

DESCRIPTION

This use case reconciles complete sets of detector data[11] in the Pre-Staging database with the Staging database. The latest loop detector location and reading information is maintained in the Pre-Staging database and is not reconciled since new detector reading data is still actively being assigned to the loop detectors.

PRECONDITON(S)

None

INITIATION OF USE CASE

This use case is invoked from the "Managing Pre-Staging Database to Staging Database" use case.

BASIC COURSE

1. Connect to the Pre-Staging database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course*.
2. Connect to the Staging database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course*.
3. Retrieve all the complete loop detector datasets from the Pre-Staging database.
4. Get the oldest loop detector dataset.
5. Get one loop detector location object and its associated reading information from the selected detector dataset.
   a. If a loop detector location object is new or any of the spatial location value for an existing loop detector location object has changed (e.g. coordinates or distance from offset values) RTMIS coordinates need to be (re)established for the loop detector location objects. Call the "Calculating Detector Location Coordinates by LRS" use case for this purpose.
   b. If no representation was found in the Staging database add this loop detector location object to the Staging database. Flag the loop detector location object in the Staging database as NEW.

[11] All detectors reading data in the pre-staging database is assigned to the latest downloaded detector location objects. This assignment continues until a new set of location information objects are downloaded into the pre-staging database. A collection of location and associated reading objects known as detector dataset is considered complete once a newer set of location objects has been downloaded. A new detector dataset is created with the newer set of location objects and subsequent readings are assigned to the location objects in the new detector dataset.

c. If representation was found, compare each loop detector location object's property with the Staging database representation and make sure the matching property values are identical. If one or more property value differs, mark the Staging database's representation as UPDATED. Update the Staging database representation where values differ.
d. If representation was found and all values are the same no action is required.
e. In all cases, change the "validate date" property of the Staging database to the system's current time.
f. Compare all the readings associated with the selected loop detector location object with the readings of the matching loop detector location object in the Staging database. If a particular reading for a given loop detector location object is not represented in the Staging database add this reading to the Staging database.

6. If there are any more unprocessed loop detector location objects, return to Step 5.
7. Remove the selected loop detector dataset (acquired in step in Step 4) from the collection of loop detector datasets (acquired in Step 3).
8. Log an activity in the Staging database and commit all transactions.
9. Remove all representation of this dataset from the Pre-Staging database.
10. Log an action in the Pre-Staging database and commit all transactions in the Pre-Staging database.
11. If there is any more loop detector datasets return to Step 4.
12. Call the "Reconcile Unmatched Detector Readings" use case.

ALTERNATIVE COURSE

1. Abort

POST-CONDITION(S)

ISSUES

None

5.2.3.3. Reconciling Unmatched Detector Readings

ACTOR(S)

Automated Service

DESCRIPTION

Enables the Actor to match an unmatched loop detector reading[12] stored in the Pre-Staging database with the appropriate loop detector location stored in the Staging database.

PRECONDITON(S)

1. An unmatched loop detector reading occurred has been stored in the Pre-Staging database unmatched loop detector readings table.
2. This use case is invoked from the "Reconcile Detector Data with Staging Database" use case.

INITIATION OF USE CASE

Parent use case, "Reconcile Detector Data with Staging Database," requests the services of the Actor.

BASIC COURSE

1. Connect to Pre-Staging database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course.*
2. Connect to Staging database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course.*
3. Request the oldest unmatched loop detector reading object in the Pre-Staging database.
4. Verify that a matching loop detector location detector ID is stored in the Staging database.
   a. If a matching detector ID is found then,
      i. Add the unmatched loop detector reading object to the Staging database.
      ii. Delete the unmatched detector reading object from the Pre-Staging database.
   b. If a matching detector ID is <u>not</u> found then check the reading date of the unmatched detector reading.

---

[12] When a loop detector reading enters the system an attempt to match the reading with its appropriate loop detector location is made. As the result of loop detector locations being downloaded and updated on a less frequent basis than loop detector readings, unmatched loop detector readings will occur when new loop detector reading enter the system before a new loop detector location. The unmatched loop detector readings table in the Pre-Staging database provides a temporary holding cell for these unmatched loop detector readings until the loop detector locations enter the system and are placed in the loop detector locations table in the Staging database.

i. If the date exceeds a pre-determined amount of time, delete the unmatched detector reading object from the Pre-Staging database.

5. Return to Step 3 until all unmatched detector reading objects in the Pre-Staging database have been reviewed.
6. Close database actions.
7. Close database connections.

ALTERNATIVE COURSE

Unable to Connect to the Pre-Staging or Staging Databases:

1. Abort.

POST-CONDITION(S)

An unmatched detector reading object in the Pre-Staging database has been assigned to its appropriate detector location in the Staging database.

ISSUES

None

5.2.4. Periodically Updated Data to Staging Database

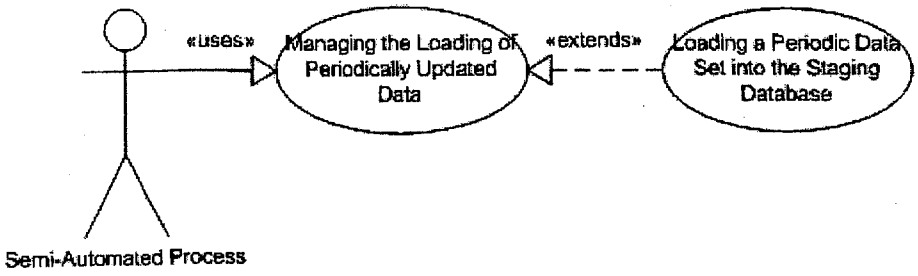
Two use cases define the process of loading periodically updated data into the Staging database. The use cases are:

1. *Managing the Loading of Periodically Updated Data*
2. *Loading a Periodic Data Set into the Staging Database*

"Managing the Loading of Periodically Updated Data" use case directs the overall process of loading periodically updated data into the Staging database. "Loading a Periodic Data Set into the Staging Database" use case extends the capability of its parent use case, "Managing the Loading of Periodically Updated Data." The child use case is responsible for loading the periodic data set into the Staging database.

Figure 9 illustrates the relationships between the "Periodically Updated Data to the Staging Database" use cases.

Figure 9: Use Cases for Periodically Updated Data to the Staging Database



5.2.4.1. Managing the Loading of Periodically[13] Updated Data into the Staging Database

ACTOR(S)

User

DESCRIPTION

Enables the User to load periodically updated data into the Staging database.

PRECONDITON(S)

1. A data set used by RTMIS has been updated by the data source.
2. SCAG acquires a periodically updated data set from the data source.
3. The data set is provided in a consistent format to the schema design housed in RTMIS.

INITIATION OF USE CASE

The User selects to load the periodically updated data set into the Staging database.

BASIC COURSE

1. The User interfaces with the desktop application that administers the loading of the periodically updated data into the Staging database.
2. The User identifies the data set and type to load.
3. The User enters basic metadata for the data entity being loaded.
4. The system identifies the target table in the Staging database.
5. The system connects to the Staging database, creates an action and logs an activity.
   a. If unable to connect to the Staging database, go to *Alternative Course*.
6. The system requests the "Loading a Periodic Data Set into the Staging Database" use case to verify and load the data set.
   a. If a replica of the data already exists in the database, then go to the *Alternative Course*.
7. The system logs an activity, closes the action and closes the connection with the Staging database.
8. The system notifies the User of the status of the database transaction.
   a. If the database transaction is successful, the system notifies the User of the successful database transaction.
   b. If the system receives an error message from a child use case, the system notifies the User of the error.
9. The User terminates the desktop application.

[13] Periodically updated data refers to data that is updated monthly, quarterly, bi-annually, annually or later.

ALTERNATIVE COURSE

Unable to Connect to the Staging database:
1. Notify User.
2. Abort.

Data Exists in the Database:
1. Notify User.
2. Abort.

POST-CONDITION(S)

The periodically updated data is loaded into the Staging database.

ISSUES

1. The data is in an inconsistent format. A document will be provided to the User specifying the pre-determined data format for each data set.

5.2.4.2. Loading a Periodic Data Set into the Staging Database

ACTOR(S)

Automated Service

DESCRIPTION

Enables the User to load the most recent version of a periodic data set, such as Caltrans FUNC, Census, or Land Use data, into RTMIS' Staging database.

PRECONDITON(S)

1. The periodic data set used by RTMIS has been updated by the agency/company.
2. SCAG acquires the dataset from the agency/company.
3. The data set is provided in a consistent format to the schema design housed in RTMIS.

INITIATION OF USE CASE

Parent use case, "Managing the Loading of Periodically Updated Data into the Staging Database", requests the services of the Actor.

BASIC COURSE

1. Verify that the data set's field format is consistent with the database table schema.
   a. If the data set's field format does not match, go to *Alternative Course.*
2. Start a database transaction.
3. Review data record's content for validity.
   a. If the data record's content is invalid, go to the *Alternative Course.*
4. Store the valid data record in the Staging database.
5. Repeat Steps 3 and 4 until no more data records exist in the dataset.
6. Commit database transaction.

ALTERNATIVE COURSE

Formatting Inconsistent with Database Schema:

1. Pass an error message to parent use case, "Managing the Loading of Periodically Updated Data into the Staging Database."
2. Abort.

Invalid Data Record:

1. Roll back database transaction.
2. Pass an error message to parent use case, "Managing the Loading of Periodically Updated Data into the Staging Database."
3. Abort.

POST-CONDITION(S)

The periodic data set update has been loaded into the Staging database.

ISSUES

None

5.2.5. Staging Database to the Production Database

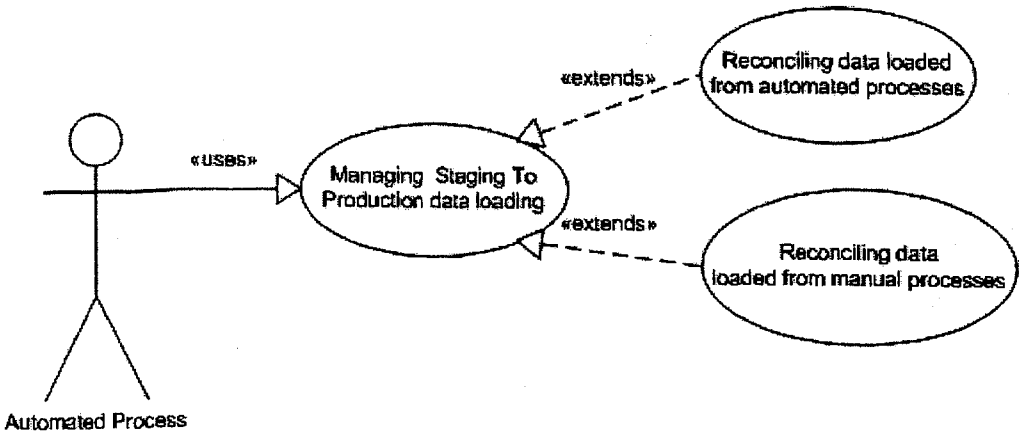
Three use cases define the process of reconciling data between the Staging Database and the Production database. The use cases are:

1. *Managing Staging to Production Data Loading*
2. *Reconciling Data Loaded from Automated Services*
3. *Reconciling Data Loaded from Semi-automated Services*

"Managing Staging to Production Data Loading" use case directs the reconciliation of new data stored in the Staging database with data in the Production database. "Reconciling Data Loaded from Automated Services" and "Reconciling Data Loaded from Semi-automated Services" use cases extend the capability of the parent use case, "Managing Staging to Production Data Loading." "Reconciling Data Loaded from Automated Services" use case is responsible for integrating the new data loaded automatically into the Staging database into the Production database. "Reconciling Data Loaded from Semi-automated Services" use case is responsible for integrating data loaded in semi-automated manner into the Staging database into the Production database.

Figure 10 illustrates the relationships between the "Staging Database to the Production Database" use cases.

Figure 10: "Staging Database to the Production Database" Use Case



5.2.5.1. Managing Staging to Production Data Loading

ACTOR(S)

Automated servic

DESCRIPTION

This use case periodically reconciles new data in the Staging database with data in the Production database. Reconciliation will occur at predetermined intervals during non-peak usage hours of the Production database.

PRECONDITON(S)

None

INITIATION OF USE CASE

This use case is invoked by an automated service that is scheduled to run at predetermined times.

BASIC COURSE

1. Connect to the Staging database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course*.
2. Connect to the Production database, create an action and log an activity.
   a. If unable to connect, go to *Alternative Course*.
3. If data loaded into the Staging database in an automated manner has not been integrated into the Production database in the past predetermined time call the "Reconciling Data Loaded from Automated Services" use case.
4. If there are any RTMIS data entities in Staging database that is loaded into the Staging database as part of a semi-automated service:
   a. Call the "Reconciling Data Loaded from Semi-Automated Services" use case.
5. Disconnect from the Production database.
6. Disconnect from the Staging database.
7. Wait for even to trigger a return to Step 1.

ALTERNATIVE COURSE

1. Disconnect from the Production database.
2. Disconnect from the Staging database.
3. Abort.

POST-CONDITION(S)

New or updated data from the Staging database has been integrated into the Production database. All data successful integrated into the Production database should either be deleted from the Staging database or flagged as updated in the Staging database.

ISSUES

None

5.2.5.2. Reconciling Data Loaded from Automated Services

ACTOR(S)

Automated Service

DESCRIPTION

This use case selects all data that was loaded into the Staging database in an automated manner and integrates it into the Production database. Currently only loop detector location and loop detector reading data is loaded into the Staging database in an automated manner. This use case itself will runs as a result of an automated service.

PRECONDITON(S)

A database connection to the Staging and Production databases has successfully been made.

INITIATION OF USE CASE

This use case's services are requested by "Managing Staging to Production Data Loading" use case.

BASIC COURSE

1. Create and log an activity with the Staging database.
2. Start a database transaction to the Staging database.
3. For each element in the loop detector data set:
   a. If an error occurs, go to *Alternative Course*.
   b. Check the 'validated date' property on the element.
      i. If the date difference between this date and the current system date more than a pre-determined time, flag this element as TO_DELETE.
   c. Log an activity with the Staging database.
4. Commit the database transaction to the Staging database.
5. Retrieve all the complete loop detector and associated reading data into a single data sets[14] from the Staging database where either or both of the following is true:
   a. The loop detector location data element is NEW, TO_DELETE or UPDATED.
   b. Where readings exist for a loop detector location object.
6. If there are one or more elements in the loop detector data set:
   a. Start a database transaction to the Production database.
   b. Start a database transaction to the Staging database.
   c. For each element in the selected loop detector data set do the following:
      i. If an error occurs, go to Alternative Course.

[14] Loop detector data sets consist of loop detected location data and loop detected reading data.

ii. If the loop detector location data is flagged as UPDATED or TO_DELETED, retire this now outdated loop detector in the Production database.
iii. If the loop detector location is flagged as TO_DELETE, alter the flag in the Staging database to DELETED.
iv. If the loop detector location is flagged as NEW or UPDATED, add the loop detector location to the Production database as the active loop detector location.
v. If the loop detector element in the Staging database has a flag remove the flag.
vi. If the loop detector element has any reading associated with it, integrate these reading into the Production database.
vii. Delete all readings from this loop detector data set element in the Staging database.

d. Commit the transaction to the Production database.
e. Commit the transaction to the Staging database.
f. Log an action in the Staging database.

ALTERNATIVE COURSE

1. If a transaction is open to the Staging database, undo all edit and close the transaction.
2. If a transaction is open to the Production database, undo all edit and close the transaction.
3. Abort.

POST-CONDITION(S)

All datasets that were loaded into the Staging database in an automated manner has been integrated into the Production database. All Staging database entities that have been integrated into the Production database have either been removed from the Staging database or the flags' flagging them as ready to be integrated into the Production database has been removed.

ISSUES

None

5.2.5.3. Reconciling Data Loaded from Semi-Automated Services

ACTOR(S)

Automated Service

DESCRIPTION

This use case selects all data that was loaded into the Staging database in a semi-automated manner and integrates it into the Production database. This use case itself will run as a result of an automated service.

PRECONDITON(S)

A database connection to the Staging and Production databases has successfully been made.

INITIATION OF USE CASE

This use case's services are requested by "Managing Staging to Production data loading" use case.

BASIC COURSE

1. Create and log an activity with the Staging database.
2. For each data set loaded in a semi-automated manner in the RTMIS Staging database:
    a. If an error occurs, go to the *Alternative Course*.
    b. Start a database transaction with the Production database.
    c. Start a database transaction with the Staging database.
    d. Establish if this RTMIS data set and version has not already been loaded into the Production database.
    e. Log an activity with the Staging database.
    f. Commit and start a database transaction with the Staging database.
    g. If the data set was not loaded or if the user requested a reload:
        i. Integrate the dataset into the Production database.
        ii. Log an activity with the Staging database.
        iii. Remove the RTMIS data set from the Staging database.
        iv. Log an activity with the Staging database.
    b. If the data set was loaded and if the user did not request a reload:
        i. Log an activity with the Staging database.
    c. Commit the database transaction to the Production database.
    d. Commit the database transaction to the Staging database.

ALTERNATIVE COURSE

1. If a transaction is open to the Staging database, undo all edits and close the transaction.
2. If a transaction is open to the Production database, undo all edits and close the transaction.
3. Abort.

POST-CONDITION(S)

All data sets that were loaded into the Staging database in a semi-automated manner have been integrated into the Production database. All Staging database entities that have been integrated into the Production database have been removed from the Staging database.

ISSUES

None

5.3. RTMIS Administrative Tools

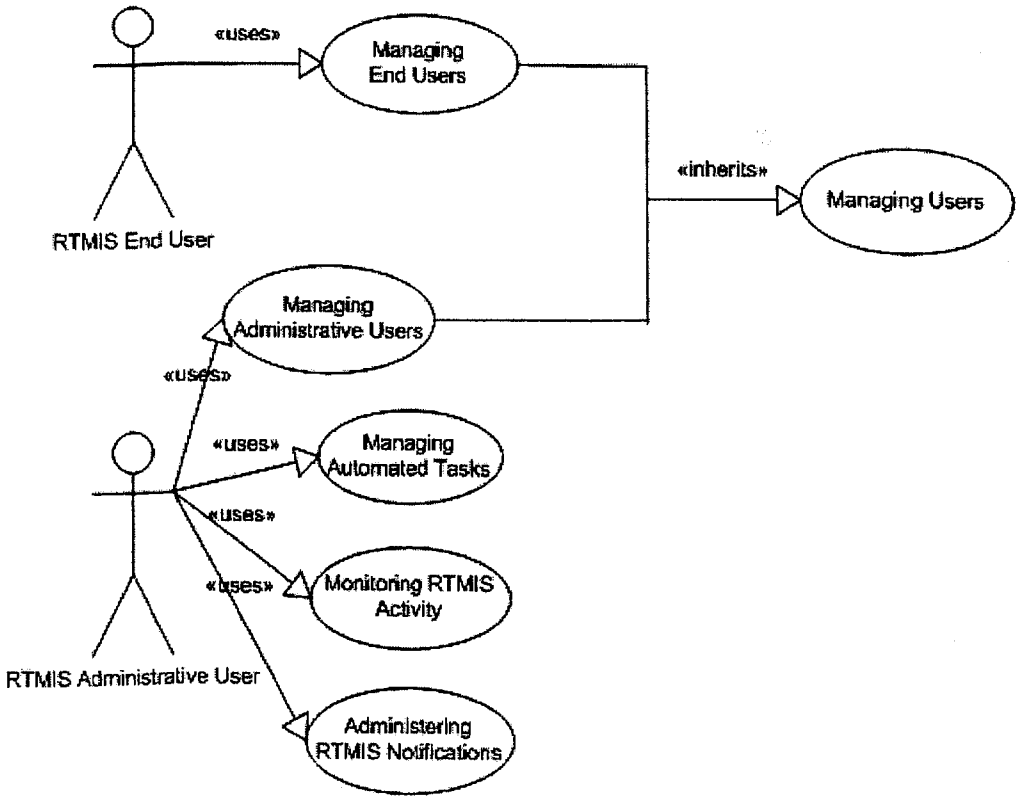
Six use cases define the RTMIS administrative tools. These use cases are:

1. *Managing End Users*
2. *Managing Administrative Users*
3. *Managing Users*
4. *Managing Automated Tasks*
5. *Monitoring RTMIS Activity*
6. *Administering RTMIS Notifications*

"Managing End Users", "Managing Administrative Users" and "Managing Users" use cases facilitate the administration of End User and Administrative User RTMIS accounts, such as adding, editing or deleting an account. "Managing Automated Tasks" use case provides the Administrative User the ability to modify an automated task's properties. "Monitoring RTMIS Activity" use case provides an Administrative User access to monitor the RTMIS activities. "Administering RTMIS Notifications" use case provides an Administrative User the ability to add, edit or update the recipients of the RTMIS notification system. RTMIS activities include automated services, both managing and monitoring, End User use and system performance.

Figure 11 illustrates the relationships between the RTMIS End User and Administrative User and the use cases.

Figure 11: "RTMIS Administrative Tools" Use Cases



5.3.1.1. Managing Users

ACTOR(S)

None

DESCRIPTION

A super user-case that is used by a child use case ("*Managing End Users*" or "*Managing Administrative Users*") to Manage RTMIS users.

PRECONDITON(S)

A valid child use case inherits functionality from this use case.

INITIATION OF USE CASE

A valid child use case inherits this use case.

BASIC COURSE

1. Using this use case the user can elect to:
    a. Add a user.
    b. Delete a user.
    c. Edit a user profile.

ALTERNATIVE COURSE

1. Abort parent and child use case

POST-CONDITION(S)

This use case successful provided functionality to its child use case by inheritance.

ISSUES

None

5.3.1.2. Managing End Users

ACTOR(S)

RTMIS End User

DESCRIPTION

This use case along with its parent use case, "*Managing Users*", facilitate administration of their end user account.

PRECONDITON(S)

The parent use case, "*Managing Users*" functionality was successfully inherited.

INITIATION OF USE CASE

A RTMIS End user requests the service of this use case.

BASIC COURSE

1. Using this use case the end user can elects to:
    a. Add an end user.
    b. Log into a user account.
        1. Delete the user account and profile logged into.
        2. Exit system.
    c. Log into a user account.
        1. Edit the user account profile logged into.
        2. Exit system.

ALTERNATIVE COURSE

1. Abort part and child use case.

POST-CONDITION(S)

The end user was able to manage their user account.

ISSUES

None

5.3.1.3. Managing Administrative Users

ACTOR(S)

None

DESCRIPTION

This use case along with its parent use case, "*Managing Users*", facilitate administration administrative user accounts.

PRECONDITON(S)

The parent use case, "*Managing Users*" functionality was successfully inherited.

INITIATION OF USE CASE

A RTMIS Administrative user requests the service of this use case.

BASIC COURSE

1. Using this use case the administrative user can elects to:
    a. Add an administrative user account and profile.
    b. Select an administrative user account.
        i. Delete the selected administrative user account and profile.
    c. Select an administrative user account.
        i. Edit the selected administrative user account's profile.
    d. Select an end user Account.
        i. Suspend the selected end user account.
    e. Select an end user account.
        i. Delete the selected end user account and profile.

ALTERNATIVE COURSE

1. Abort part and child use case

POST-CONDITION(S)

The administrative user was able to manage any administrative user account.

ISSUES

None

5.3.1.4. Managing Automated Tasks

ACTOR(S)

Administrative User

DESCRIPTION

Enables the Administrative User to manage the properties of an automated task.

PRE-CONDITIONS

1. An Administrative User with appropriate privileges has successfully logged into the system.

INITIATION OF USE CASE

RTMIS Administrative User requests to edit the properties of an automated task.

BASIC COURSE

1. The Administrative User selects an automated task.
2. The Administrative User edits one or many properties of the automated task.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

An automated task's properties have been modified by an Administrative User.

ISSUES

None

5.3.1.5. Monitoring RTMIS Activity

ACTOR(S)

Administrative User

DESCRIPTION

Enables the RTMIS Administrator User to monitor RTMIS activities, including all automated services, user activity and system performance.

PRE-CONDITION(S)

1. An administrative User with appropriate privileges has successfully logged into the system.

INITIATION OF USE CASE

The Administrative user requests the services of this use case.

BASIC COURSE

1. The Administrative User elects to monitor the automated services of RTMIS.
    a. The Administrative User specifies the activity type of interest.
    b. The Administrative User specifies the individual activity of interest.
    c. The Administrative User specifies the period of performance of interest for the activity.

ALTERNATIVE COURSE(S)

None

POST-CONDITION(S)

The Administrative User is aware of RTMIS activities.

ISSUES

None

5.3.1.6. Administering RTMIS Notifications

ACTOR(S)

Administrative User

DESCRIPTION

Enables the RTMIS Administrator User to monitor RTMIS activities, including all automated services, user activity and system performance.

PRE-CONDITION(S)

1. An administrative User with appropriate privileges has successfully logged into the system.
2. An individual has submitted a request to the Administrative User to add him/her as a RTMIS notification member, delete him/her from the notification log or change his/her RTMIS notification settings.

INITIATION OF USE CASE

The Administrative user requests the services of this use case.

BASIC COURSE

1. If an individual would like to be added to the RTMIS notification system, then
    a. The Administrative User creates a new member profile in the notification system.
    b. The Administrative User enters the new member's contact information into the profile.
    c. The Administrative User sets the member's notification settings.
1. If an individual would like his/her profile deleted from the RTMIS notification system, then the Administrative User deletes the member's profile from the notification system.
2. If an individual would like to have his/her notification settings updated, then
    a. The Administrative User updates the member's notification settings.

ALTERNATIVE COURSE(S)

None

POST-CONDITION(S)

A new member has been added to the notification system.

ISSUES

None

5.4. RTMIS Analysis and Reporting Tools

Presently, 15 use cases define the RTMIS End User Analysis and Reporting Tools. These use cases are:

End User Operations Use Cases:

1. *Comparing Database Entity Elements*
2. *Comparing Individual Elements*
3. *Comparing Element Groups*
4. *Performing Statistical Analysis*
5. *Historic Trend Analysis*
6. *Metadata Reporter*
7. *Displaying Results*
8. *Displaying Results as a Map*
9. *Displaying Results as a Graph*
10. *Displaying Results as a Report*

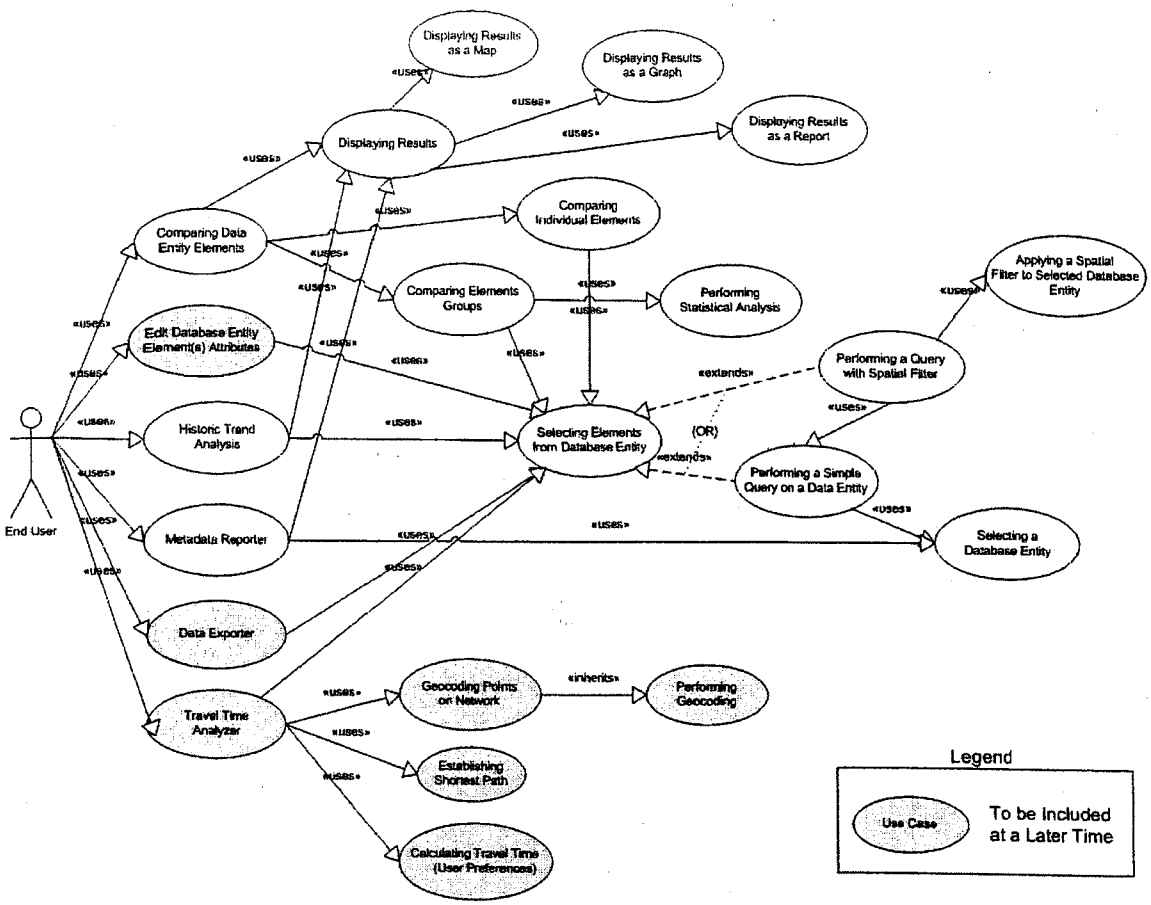
Querying the RTMIS Data Warehouse Use Cases:

11. *Selecting Elements from a Database Entity*
12. *Performing a Query with a Spatial Filter*
13. *Applying a Spatial Filter to Selected Database Entity*
14. *Performing a Simple Query on a Database Entity*
15. *Selecting a Database Entity*

"Selecting Elements form a Database Entity", "Performing a Query with a Spatial Filter", "Applying a Spatial Filter to a Selected Database Entity", "Performing a Simple Query on a Database Entity" and "Selecting a Database Entity" use cases are responsible for acquiring the database entity and/or database entity elements. This data is utilized by the End User operations, such as "Historic Trend Analysis" and "Comparing Database Entity Elements."

Figure 12 (followed on the next page) illustrates the relationships between the End User and the use cases. Some use cases included in the figure will be completed after the initial prototype of the system has been delivered to SCAG.

Figure 12: Use Case for End User Interaction



5.4.1.1. Comparing Database Entity Elements

ACTOR(S)

End User

DESCRIPTION

Enables the End User to compare elements within a database entity.

PRE-CONDITION(S)

None

INITITATION OF USE CASE

The End User elects to compare elements within a database entity.

BASIC COURSE

1. The End User connects to the RTMIS data warehouse, creates an action and logs an activity.
   a. If unable to connect to the database, go to the *Alternative Course*.
2. If the End User elects to compare two individual elements of a database entity, then
   a. "Selecting Elements from Database Entity" use case is called.
3. If the End User elects to compare two group elements of a database entity, then
   a. "Selecting Elements from Database Entity" use case is called.
4. The End User closes the action and disconnects from the RTMIS data warehouse.

ALTERNATIVE COURSE(S)

1. Notify the End User.
2. Abort.

POST-CONDITION(S)

The End User has selected the type of comparison to perform on the database entity.

ISSUES

None

5.4.1.2. Comparing Individual Elements

ACTOR(S)

End User

DESCRIPTION

Enables the End User to perform a comparison between two individual elements in a database entity.

PRE-CONDITION(S)

This use case is called by the parent use, "Comparing Elements in a Database Entity."

INITIATION OF USE CASE

The End User elects to perform a comparison between two individual elements in a database entity.

BASIC COURSE

1. The End User acquires its selected database entity by calling the "Selecting Elements from a Database Entity" use case.
2. The End User acquires its first selected by calling the "Selecting Elements from a Database Entity" use case.
3. The End User acquires its second selected element by calling the "Selecting Elements from a Database Entity" use case.
4. The End User determines the difference between the elements.
5. The End User acquires the displayed comparison of two elements by calling the "Display Results" use case.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User had compared two individual elements from a database entity.

ISSUES

None

5.4.1.3. Comparing Element Groups

ACTOR(S)

End User

DESCRIPTION

Enables the End User to perform a comparison between element groups in a database entity.

PRE-CONDITION(S)

This use case is called by the parent use, "Comparing Elements in a Database Entity."

INTIATION OF USE CASE

The End User elects to perform a comparison between element groups in a database entity.

BASIC COURSE

1. The End User acquires the selected database entity by calling the "Selecting Elements from a Database Entity" use case.
2. The End User specifies the group analysis type.
3. The End User acquires the first element group by calling the "Selecting Elements from a Database Entity" use case.
4. The End User performs the selected analysis type on the first element group by calling the "Performing Statistical Analysis" use case.
5. The End User acquires the second element group by calling the "Selecting Elements from a Database Entity" use case.
6. The End User performs the selected analysis type on the second element group by calling the "Performing Statistical Analysis" use case.
7. The End User compares the element groups' analysis results.
8. The End User acquires a display of the comparison by calling the "Display Results" use case.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User has compared two element groups.

ISSUES

None

5.4.1.4. Performing Statistical Analysis

ACTOR(S)

End User

DECRIPTION

Enables the End User to perform statistical analysis on the element groups selected for comparison. A mean is calculated to describe the elements as a whole in each element group.

PRE-CONDITION(S)

1. The End User has acquired an element group.
2. The collection of elements for the element group is provided.

INITIATION OF USE CASE

The "Comparing Element Groups" use case has requested the services of this use case.

BASIC COURSE

1. An End User specified statistical operation is performed on the element group.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

A single value has been computed to represent elements in each element group.

ISSUES

None

5.4.2.1. Historical Trend Analysis

ACTOR(S)

End User

DESCRIPTION

Enables the End User to perform historical trend analysis for a database entity element.

PRE-CONDITION(S)

None

INITIATION OF USE CASE

The End User elects to perform historical trend analysis on a database entity element.

BASIC COURSE

1. The End User connects to the RTMIS data warehouse, creates an action and logs an activity.
    a. If unable to connect to the database, go to the *Alternative Course*.
2. The End User retrieves a collection of objects form one database entity by calling the "Selecting Elements from a Database Entity" and the "Selecting Elements from a Database Entity" use cases.
3. The End User specifies an attribute for the collection of elements by calling the "Selecting Elements from a Database Entity" use case.
4. The End User acquires the results, a collection of objects sorted by time, from the selected temporal attribute filter by calling the "Selecting Elements from a Database Entity" use case.
5. The End User closes the action and disconnects from the RTMIS data warehouse.
6. The End User acquires a "change over time" display for the results by calling "Display Results" use case.

ALTERNATIVE COURSE

1. Notify the End User.
2. Abort.

POST-CONDITION(S)

The End User has acquired a historical trend analysis for a database entity element.

ISSUES

None

5.4.3.1 Metadata Reporter

ACTOR(S)

End User

DESCRIPTION

Enables the End User to acquire the associated metadata for a RTMIS database entity.

PRE-CONDITION(S)

1. Metadata has been created for all RTMIS database entities.

INITIATION OF USE CASE

The End User requests to acquire the metadata for a RTMIS database entity.

BASIC COURSE

1. The End User connects to the RTMIS data warehouse, creates an action and logs an activity.
    a. If unable to connect to the database, go to the *Alternative Course*.
2. The End User selects a database entity by calling the "Selecting a Database Entity" use case.
3. The metadata associated with database entity is obtained.
4. The End User closes the action and disconnects from the RTMIS data warehouse.
5. The End User acquires a metadata report by calling the "Display End User Results" use case.

ALTERNATIVE COURSE

1. Notify the End User.
2. Abort.

POST-CONDITION(S)

The End User acquires a metadata report for a selected database entity.

ISSUES

None

5.4.4.1. Selecting Elements from the Database Entity

ACTOR(S)

End User

DESCRIPTION

Enables the End User to select elements from a database entity.

PRE-CONDITION(S)

1. The End User elects to perform one of the following End User operations:
    a. Compare Elements in a Database Entity
    b. Analyze Historic Data
    c. Create a Metadata Report
    d. Export Data
    e. Edit Database Entity Element Attributes[15]
    f. Analyze Travel Time[16]
2. The End User is connected to the RTMIS data warehouse.

INITIATION OF USE CASE

The End User elects to select elements from a database entity.

BASIC COURSE

1. The End User submits a request to select elements from a database entity(ies).
2. If the End User elects to perform a query with a spatial filter, then the system requests the "Performing a Query with a Spatial Filter" use case.
3. If the End User elects to perform a simple query, then the system requests the "Performing a Simple Query on a Database Entity" use case.

ALTERNATIVE COURSE

Unable to Connect to Data Warehouse:
1. Notify the End User.
2. Abort.

POST-CONDITION(S)

The End User selects either a spatial query filter or a simple attribute query filter. The system begins the selected process and returns the results to the End User.

ISSUES

None

[15, 18] Envisioned future capability.

5.4.4.2. Performing a Query with a Spatial Filter

ACTOR(S)

End User

DESCRIPTION(S)

Enables the End User to perform a query with a spatial filter on a database entity. A query with a spatial filter is a query where a primary database entity is spatially filtered by a secondary database entity or multiple secondary database entities. The spatial filter can find all primary database entity features that are "contained within," "within a distance of" or "intersects" the secondary database entity(ies).

PRE-CONDITION(S)

1. The system is connected to the RTMIS data warehouse.

INITIATION OF USE CASE

The End User elects to select elements from a database entity through a spatial query.

BASIC COURSE

1. The End User selects the primary database entity to be spatially filtered.
   a. If the End User elects, an attribute query is applied to the primary database entity.
2. The End User selects a secondary database entity with a spatial dimension that will act as the primary database entity's spatial filter.
   a. If the End User elects, an attribute query is applied to the primary database entity.
3. If the End User elects, additional database entities with a spatial component are selected.
   a. Repeat Step 2 until all spatial filters have been selected by the End User.
4. The End User applies the spatial filter(s), secondary database entity(ies), to the primary database entity by calling the "Applying a Spatial Filter to a Selected Database Entity" use case.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User has performed a query with a spatial filter.

ISSUES

None

5.4.4.3. Applying a Spatial Filter to Selected Database Entity

ACTOR(S)

End User

DESCRIPTION

Enables the End User to apply a spatial filter to a selected database entity.

PRE-CONDITION(S)

1. The system is connected to the RTMIS data warehouse.
2. The End User has selected a database entity.

INITIATION OF USE CASE

The End User elects to apply a spatial filter to the data entity.

BASIC COURSE

1. The End User selects the type of spatial filter to be applied to the database entity.
   a. The End User selects a "contained within" spatial filter type.
   b. The End User selects a "within a distance of" spatial filter type.
   c. The End User selects a "intersects" spatial filter type.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

A query with a spatial filter has been applied to a database entity.

ISSUES

<u>Example of Applying a Spatial Filter to a Database Entity:</u>

1. The End User elects to find all loop detectors on the 405 freeway that are contained or within a 2 mile distance of the City of Tustin.

   Primary Database Entity: Loop Detector Locations
   Primary Database Applied Attribute Query: 405 Freeway
   Secondary Database Entity: Cities
   Secondary Database Applied Attribute Query: Tustin
   Spatial Filter Type 1: "contained"
   Spatial Filter Type 2: "within a distance of"

5.4.4.4. Performing a Simple Query on a Data Entity

ACTOR(S)

End User

DESCRIPTION

Enables the End User to perform a simple query on a database entity. A simple query applies an attribute or temporal attribute filter to a database entity.

PRE-CONDITION(S)

1. The system is connected to the RTMIS data warehouse.

INITIATION OF USE CASE

The End User elects to select elements from a database entity.

BASIC COURSE

1. The End User selects the database entity to be filtered by calling "Selecting a Database Entity" use case.
    a. If the End User elects to filter the selected database entity elements by attribute then one or more attribute filters are applied.
    b. If the End User elects to filter by temporal attribute then one or more temporal attribute filters are applied.
    c. If End User elects to not apply an attribute or temporal filter then all data elements over time for the database entity are returned.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User has filtered a database entity's elements by attribute or time.

ISSUES

Examples of Applying a Filter to a Database Entity

1. Simple Attribute Query:
The end user would like to query the Loop Detector Locations data entity for all loop detectors located along the 405 Freeway.
Data Entity: Loop Detector Locations
Attribute Query Filter: 405 Freeway
2. Simple Attribute Query w/ Temporal Component:
The end user would like to query the Loop Detector Readings data entity for all readings taken during August 2002.
Data Entity: Loop Detector Locations
Temporal Query Filter: August 2002

3. Simple Multiple Attribute Query w/ Temporal Component:
The end user would like to query the Loop Detector Readings data entity for all readings taken during August 2002 along the 405 Freeway with a traffic volume greater than *n*.
Data Entity: Loop Detector Locations
Attribute Query Filter: 405 Freeway and Traffic Volume
Temporal Query Filter: August 2002
4. Simple Multiple Attribute Query w/ Multiple Temporal Components:
The end user would like to query the Loop Detector Readings data entity for all readings taken during August 2001 and August 2002 along the 405 freeway with a traffic volume greater than *n*.
Data Entity: Loop Detector Locations
Attribute Query Filter: 405 Freeway and Traffic Volume
Temporal Query Filter: August 2001 and August 2002

5.4.4.5. Selecting a Database Entity

ACTOR(S)

End User

DESCRIPTION

Enables the End User to select a database entity.

PRE-CONDITION(S)

1. The system is connected to the RTMIS data warehouse.

INITIATION OF USE CASE

The End User needs to select a database entity.

BASIC COURSE

1. The End User selects a RTMIS database entity.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

A database entity has been selected by the End User for performing its simple query or query with a spatial filter.

ISSUES

None

5.4.5.1. Displaying End User Results

ACTOR(S)

End User

DESCRIPTION

Enables the End User to acquire a display of reporting, analysis or comparison results in the form of a map, graph or report. Certain display types may not be available for the type of operation performed. The system will only provide to the End User the appropriate display types for the operation performed. This parent use case manages the child use cases that create the data displays.

PRE-CONDITION(S)

1. The End User has successfully acquired data results from a system reporting, analysis or comparison operation.

INITIATION OF USE CASE

One of the End User's reporting, analysis or comparison use cases requests the services of this use case.

BASIC COURSE

1. The system provides to the End User the selection of available display options for the reporting, analysis or comparison operation performed.
2. If the map display option is available and the End User elects to acquire the data results in map form, then the "Displaying Results in a Map" use case is called.
3. If the graph display selection is available and the End User elects to acquire the results in graph form, then the "Displaying Results in a Map" use case is called.
4. If the report display option is available and the End User elects to acquire the results in report form, then the "Displaying Results in a Report" use case is called.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The reporting, analysis or comparison data results are provided to the End User in the form of a map, graph or report.

ISSUES

None

5.4.5.2. Displaying Results as a Map

ACTOR(S)

End User

DESCRIPTION

Enables the End User to acquire a display of analysis/comparison results in the form of a map. This use case extends the capability of the parent use case, "Displaying Results."

PRE-CONDITION(S)

1. The End User elects to have the results of a reporting, analysis or comparison operation displayed in the form of a map.
2. The results of the reporting, analysis or comparison operation are provided.

INITIATION OF USE CASE

The parent use case, "Displaying Results", requests the services of this use case.

BASIC COURSE

1. The results of the End User's operation are rendered on a map with appropriate database entities rendered in the background.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User is provided a map presenting the results of the End User's operation.

ISSUES

None

5.4.5.3. Displaying Results as a Graph

ACTOR(S)

End User

DESCRIPTION

Enables the End User to acquire a display of analysis/comparison results in the form of a graph. This use case extends the capability of the parent use case, "Displaying Results."

PRE-CONDITION(S)

1. The End User elects to have the results of a reporting, analysis or comparison operation displayed in the form of a graph.
2. The results of the reporting, analysis or comparison operation are provided.

INITIATION OF USE CASE

The parent use case, "Displaying Results", requests the services of this use case.

BASIC COURSE

1. The results of End User's operation are plotted on a graph.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User is provided a graph presenting the results of the End User's operation.

ISSUES

None

5.4.5.4. Displaying Results as a Report

ACTOR(S)

End User

DESCRIPTION

Enables the End User to acquire a display of analysis/comparison results in the form of a report. This use case extends the capability of the parent use case, "Displaying Results."

PRE-CONDITION(S)

1. The End User elects to have the results of a reporting, analysis or comparison operation displayed in the form of a report.
2. The results of the reporting, analysis or comparison operation are provided.

INITIATION OF USE CASE

The parent use case, "Displaying Results", requests the services of this use case.

BASIC COURSE

1. The results of End User's operation are documented in a report.

ALTERNATIVE COURSE

None

POST-CONDITION(S)

The End User is provided a report stating the results of the End User's operation.

ISSUES

None

6. RECOVERY AND FALLBACK STRATEGY

As with any enterprise system, RTMIS must be recoverable from unexpected disasters or machine failure. RTMIS itself will not implement functionality to manage this backup process, but procedures need to be in place to facilitate system recovery. Four RTMIS data sets can be identified for backup:

- Physical files acquired from the Internet residing on the data acquisition workstation.
- Data in the Pre-Staging database.
- Data in the Staging database.
- Data in the Production database.

The bulk of the data will reside in the Production database. The other three data sources listed above would in terms of data volume only be a fraction of the size of the Production database.

A detailed RTMIS recovery and fallback strategy will be determined with the SCAG staff. The following early recommendations can however be made:

1. In order to minimize the loss of data from live data sources, computers and associated services should be restarted in the following order:
    a. Data acquisition workstation and data acquisition services.
    b. Production database server, RTMIS database services and associated RTMIS user components.
    c. Web server and RTMIS web services.
    d. Pre-Staging and Staging databases services on the database server.
    e. Automated database loader workstation and associated services.
2. Should all the RTMIS data sources have to be recovered and once step 1a. above has been completed, these data sources should be recovered in the following order:
    a. Production database
    b. Staging database
    c. Pre-Staging database
    d. Physical files on the data acquisition workstation.

RTMIS data should be backed up daily. To minimize the impact that backup will have on the RTMIS system, backups should be performed during off-peak hours. An early recommendation for a backup schedule and backup types is summarized in Table 6.1 below:

Table 6.1: RTMIS preliminary backup schedule.

	Mon.	Tue.	Wed.	Thur.	Fri.	Sat.	Sun.
Physical Files	F	F	F	F	F	F	F
Pre-Staging	F	F	F	F	F	F	F
Staging	F	F	F	F	F	F	F
Production	I	I	I	I	I	I	I & F
Legend: F - Full backup I - Incremental Backup							

APPENDIX A: GLOSSARY OF TERMS

24/7 (Pronounced 'twenty –four seven') – Twenty-four hours a day, seven days a week.

Census Block – The smallest geographic area for which the Census Bureau collects and tabulates census data. Block boundaries are formed by streets, roads, railroads, streams, and other bodies of water, as well as other visible physical and cultural features, such as legal boundaries. Many Census Blocks compose a Census Block Group; many Census Block Groups compose a Census Tract. See also Census Tract.

Census Place – Concentration of Population

Census Tract – A small, relatively permanent geographic entity within county delineated by a committee of local data users. Tracts have between 2500-8000 residents and boundaries that follow visible features. See also Census Blocks.

Data Warehouse – A collection of data, through the combination of different databases across an enterprise, designed to support decision making.

ESRI – A GIS software development company located in Redlands, CA.

ESRI's ArcIMS (Internet Mapping Service) – A software application that enables the distribution of high-end geographic information systems and mapping services via the Internet.

ESRI's ArcObjects – A collection of software components with GIS functionality and programmable interfaces. These software components, based on COM protocol, are the framework of ESRI's ArcGIS Desktop Application.

ESRI's ArcSDE (Spatial Database Engine) – A software application that extends the capability of several industry-standard relational database management systems (RDBMS) in storing spatial data. The ArcSDE server works cooperatively with the RDBMS server to store and retrieve spatial data. The physical storage for the data, the relational table, is provided by the RDBMS while the geometric data interpretation for a GIS is handled by ArcSDE.

ArcSDE software provides a number of geographic analysis tools that software developers can use to process data requested from the server.

ESRI's Coverage - A vector file format for storing data storage the location, shape, and attributes of geographic features. Map features are stored as both primary (arcs, polygons, and points) and secondary features (tics, links, and annotation). The attributes of the geographic features are stored independently in feature attribute tables.

ESRI's Shapefile – A vector file format for storing the location, shape and attributes of geographic features. It is stored in a set of related files and contains one feature class.

FUNC – Functionally Classified Roads

Georeference – To assign coordinates from a known reference system, such as latitude/longitude, State Plane or UTM, to the page coordinates of an image or a map.

GIS – Geographic Information Systems

HPMS (Highway Performance Monitoring System) – A nation-wide effort by the States to inventory of the nation's interstate and state highway characteristics and assets.

LRS (Linear Referencing System) – A system that utilizes relative positions, measures, to identify locations along linear features. For example, Route 405, mile 22.

MPAH – Master Plan of Arterial Highways

OCTA – Orange County Transportation Authority

OLAP (Online Analytical Processing) – A category of software tools that provide dimensional analysis of multi-dimensional data stored in a database. Examples of dimensional analysis are time series and trend views.

OLTP (Online Transaction Processing) – A type of computer processing in which each request is individually registered as a transaction. Transaction processing requires interaction with a user, such an automated servic.

PeMS - Performance Monitoring System

Pre-Staging Database - One of the three RTMIS RDBMS. This database's main purpose is to provide the initial load of the continuously collected data into the RTMIS database schema.

Production Database – One of the three RTMIS RDBMS. This database supports RTMIS use operations. This database is also referred to as the RTMIS data warehouse.

RTMIS – Regional Transportation Monitoring Information System

Staging Database - One of the three RTMIS RDBMS. This database's main purposes are to standardize measures amongst the data entities, to optimize the system by pre-processing anticipated data queries and to ensure data meets production standards before loading the data into the Production database.

TANN – Travel Advisory News Network

TBM – Thomas Brothers Maps

RTMIS System Hardware Requirements

Database server:

PowerEdge 4600: PowerEdge 4600,Intel Xeon 2.2GHZ/512K Cache
Additional Processors: Free! 2nd Processor,Intel® Xeon 2.2GHZ/512K Cache
Memory: 2GB DDR SDRAM 4X512MB
Keyboard: Standard Windows Keyboard,Gray
Monitors: 17in (16in Viewable) Monitor Gray,P793
1st Hard Drive: 73GB 10K RPM Ultra 160 SCSI Hard Drive 73GB
Diskette Drive: 3.5in,1.44MB FLOPPY DRIVE
Operating Systems: Windows 2000 Server (with appropriate licensing)
Mouse: Logitec System Mouse,Gray
CDROM or DVDROM: 24X IDE CD-ROM
Hard Drive Backplane: 1X8 Hot-Pluggable Hard Drive Backplane
Documentation: Electronic Documentation
2nd Hard Drive: 73GB 10K RPM Ultra 160 SCSI Hard Drive
Hard Drive Configuration: Embedded RAID 5,No Drives in Media Bay (Min of 3 drives required)
Chassis Orientation: Tower Orientation
Hardware Support Services: 3Yr Same Day 4Hr Response Parts + Onsite Labor (M-F 8am-6pm)
Installation Support Services: No Installation
4th Hard Drive: 73GB,10K RPM,1in Ultra3 (Ultra 160) SCSI Hot-Plug Hard Drive
5th Hard Drive: 73GB,10K RPM,1in Ultra3 (Ultra 160) SCSI Hot-Plug Hard Drive
3rd Hard Drive: 73GB,10K RPM,1in Ultra3 (Ultra 160) SCSI Hot-Plug Hard Drive
Uninterruptible Power Supply: 120V,2200VA,3U Rack-mount UPS with software (requires Dell 120V PDU)
Power Supply Kits: Redundant Power Supplies

This configuration does not include backup hardware and software. It is presumed that SCAG has backup capabilities that would be adequate for this database server.

PowerEdge 4600
Estimated Total Price: $9,205 – $12,150

Web server:

Base: PowerEdge 2600, Intel Xeon 2.4GHz/512K Cache
Additional Processors: Single Processor Only
Memory: 2GB DDR SDRAM (4X512MB)
Keyboard: Standard Windows Keyboard, Gray
Monitors: Dell E772, 17 inch (16.0 inch Viewable), Gray Monitor
1st Hard Drive: 36GB 10K RPM Ultra 320 SCSI Hard Drive
Floppy: 3.5 in, 1.44MB, Floppy Drive
Operating System: Windows 2000 Server with appropriate license
Mouse: Logitec Mouse
CD-ROM: 24X IDE CD-ROM
BackPlanes: 1X6 Hot-Pluggable Backplane,PE2600
Documentation: Electronic Documentation,P2600
2nd Hard Drive: 36GB 10K RPM Ultra 320 SCSI Hard Drive
Factory Configurations: Drives attached to embedded SCSI controller, No RAID
Chassis Style: Tower Chassis Orientation,P2600 TOWER
Hardware Support Services: 3Yr Same Day 4Hr Response Parts + Onsite Labor (M-F 8am-6pm)

Add a high-end video card capable of generating map graphics with adequate performance.

PowerEdge 2600
Estimated Total Price: $5,610 - $7,405

Acquisition and Data Loading Workstations:

Dell Dimension 8200 Series: Pentium® 4 Processor at 2.40GHz with 533MHz system bus/ 512K L2 Cache D8224B
Memory: 256MB PC800 RDRAM 256M82
Keyboard: Dell® Quietkey® Keyboard QK
Monitor: New 17 in (16.0 in v.i.s., .27dp) E772 Monitor E772
Video Card: New 64MB DDR NVIDIA GeForce4™ MX Graphics Card with TV
Hard Drive: 40GB Ultra ATA/100 7200RPM Hard Drive 40
Floppy Drive: 3.5 in Floppy Drive 3
Operating System: Microsoft® Windows® 2000 Professional
Mouse: Dell® 2-button scroll mouse SM
Broadband Ready/ Ethernet Network Card: 10/100 PCI Fast Ethernet NIC CNET
Modem: No Modem Requested N
CD or DVD Drive: 40x/10x/40x Max CD-R
Sound Card: Sound Blaster Live! Digital Sound Card
Speakers: No Speaker Option N
Productivity Software: No Software Package Requested MSONO
Virus Protection: Norton AntiVirus™ 2002, 90-day introductory offer
Limited Warranty, Services and Support Options: 3 Year Limited Warranty plus 3 Year On-site Service U3OS
Installation Services: No Installation
READYWARE: No ISP Requested NISP
Internet Access Service: No ISP Requested NISP

Dell Dimension 8200 Series
Estimated Total Price: $1,350 – 1,780 each

Estimated Price for both: $2,700 – $3,560

A UPS will be required for the data acquisition workstation should SCAG not have UPS capabilities.